

PATENT ABSTRACTS OF JAPAN

(11)Publication number : 11-004270

(43)Date of publication of application : 06.01.1999

(51)Int.Cl. H04L 27/34
H03M 13/12
H03M 13/22

(21)Application number : 09-339662 (71)Applicant : KONINKL PHILIPS ELECTRON
NV

(22)Date of filing : 10.12.1997 (72)Inventor : CHOULY ANTOINE
GIANELLA DAVID

(30)Priority

Priority number : 96 9615159 Priority date : 10.12.1996 Priority country : FR

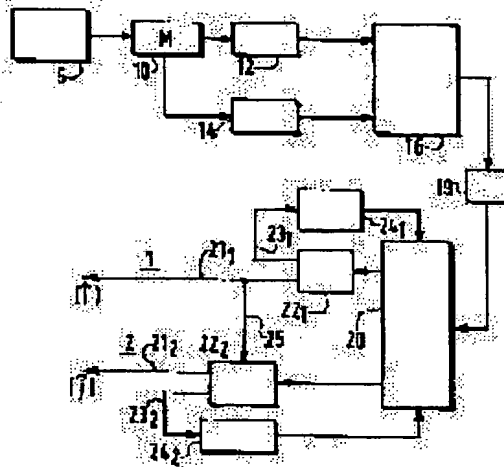
(54) DIGITAL TRANSMISSION SYSTEM AND METHOD

(57)Abstract:

PROBLEM TO BE SOLVED: To improve a performance and to make it possible for a system to properly function by the minimum S/N by combining a generating code generated by using a systematic superimposed trellis encoding with a multidimensional amplitude modulation having a dimension Q which has 2M number of states and using a block decode as a repetitive decode.

SOLUTION: Input data from a source 5 is converted to a 16-QAM symbol by a memory 10, a row encoding means 12, a column encoding means 14 and a 4-AM digital modulation symbol assignment element 16, and transmitted to a buffering means 20 through a channel 19. A viterbi decoder 221 and a parity code decoder 222 output hard discrimination outputs 211 and 212 and execute a loop backs of soft discrimination outputs 231 and 232 to the buffering means 20 by way of data interleavers 241 and 242.

Decoding by the decoder 222 is executed on the basis of the hard discrimination output 211. Thus, credibility of the hard discrimination outputs 211 and 212 can be improved as a repetitive processing repeats.



LEGAL STATUS

[Date of request for examination]

[Date of sending the examiner's decision of rejection]

[Kind of final disposal of application other than the examiner's decision of rejection or application converted registration]

[Date of final disposal for application]

[Patent number]

(19) 日本国特許庁 (J P)

(12) 公開特許公報 (A)

(11) 特許出願公開番号

特開平11-4270

(43) 公開日 平成11年(1999) 1月6日

(51) Int.Cl.⁸

識別記号

F I

H 0 4 L 27/34

H 0 4 L 27/00

E

H 0 3 M 13/12

H 0 3 M 13/12

13/22

13/22

審査請求 未請求 請求項の数7 O L (全 24 頁)

(21) 出願番号

特願平9-339662

(22) 出願日

平成9年(1997)12月10日

(31) 優先権主張番号 9 6 1 5 1 5 9

(32) 優先日 1996年12月10日

(33) 優先権主張国 フランス (F R)

(71) 出願人 590000248

コーニンクレッカ フィリップス エレク
トロニクス エヌ ヴィ

Koninklijke Philips
Electronics N. V.

オランダ国 5621 ペーアー アインドー
フェン フルーネヴァウツウェッハ 1

(72) 発明者 アントワーン、シュリー

フランス国パリ、リュ、デ、マリニエ
ール、4

(74) 代理人 弁理士 佐藤 一雄 (外3名)

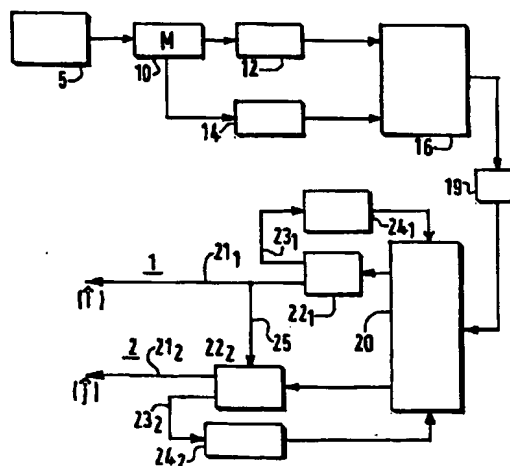
最終頁に続く

(54) 【発明の名称】 デジタル伝送システムおよび方法

(57) 【要約】

【課題】 入力データを保護するデジタル伝送システムおよび方法を提供する。

【解決手段】 デジタル伝送システムは、系統的重畳コードを入力データに適用することで、ブロック単位にて、生成コードを生成する。トレリスは、パリティコードビットから成る冗長データを加えることで、閉鎖される。次に、こうして符号化されたデータが、多次元デジタル変調されたシンボルに割当てられる。受信機端において、復号手段は、継続された2つの経路に沿って、反復的復号を遂行する。多次元デジタル変調されたシンボルの各サブセット（部分集合）に対してハード判定の信頼性が計算され、これからソフト判定が生成される。第2の経路は、第1の経路によって生成された結果を使用してソフト判定を生成する。このソフト判定が出力シンボルを決定するために用いられる。



【特許請求の範囲】

【請求項 1】入力データを保護するためにデジタル伝送システム内で実現されるデジタル伝送方法であって、前記入力データの符号化フェーズと反復的復号フェーズとを備え、

前記符号化フェーズは、前記反復的復号を可能にするための系統的重ねトレリス符号化を遂行する第 1 のステップと、前記第 1 のステップと組み合わせて用いられる、前記符号化されたデータを、デジタル的に変調されたシンボルに割当てる第 2 のステップとを含み、

前記第 1 のステップの際に、前記系統的重ねトレリス符号化が $P/(Q, M)$ なる符号化レートを持ち、ここで、 M 、 P および Q は整数であり、かつ、 Q および M は 1 より大きく、

前記系統的重ねコードは、生成コードをブロック毎に生成するために使用され、トレリスが冗長データの追加によって閉鎖され、

前記生成コードが、系統的重ねコードを介して符号化されたデータを組み合わせたマトリックスのロウ符号化およびカラム符号化によって生成された要素によって形成され、

前記第 2 のステップは、前記生成コードを、 2^M 個の状態を持つ次元 Q を持つ多次元振幅変調と結合し、前記反復的復号がブロック復号であることを特徴とする方法。

【請求項 2】前記符号化レートは、 $7/8$ に等しく、前記変調は、8 次元を有する 8-D 変調であることを特徴とする請求項 1 記載の方法。

【請求項 3】入力データの保護を有するデジタル伝送システムであって、前記データに対する符号化手段と反復的復号手段とを備え、

前記符号化手段は：

—前記反復的復号を可能にするための系統的重ねトレリス符号化を遂行するための第 1 のサブセット、および—前記符号化されたデータをデジタル的に変調されたシンボルに割当てるための第 2 のサブセットとを含み、

前記第 1 のサブセットは、 $P/(Q, M)$ なる符号化レートを有する系統的重ねトレリス符号化を利用し、ここで、 M 、 P および Q は整数であり、かつ、 Q および M は 1 より大きく、

前記系統的重ねコードは、生成コードをブロック毎に生成するために使用され、少なくとも 1 つのパリティコードビットを含む冗長データの追加によってトレリスが閉鎖され、

前記生成コードは、系統的重ねコードを介して符号化されたデータを組み合わせるマトリックスのロウ符号化およびカラム符号化によって生成された要素によって形成され、

前記第 2 のサブセットは、前記生成コードを、 2^M 個の状態を持つ次元 Q を有する多次元振幅変調と結合し、

前記反復的復号は、ブロック復号であることを特徴とするデジタル伝送システム。

【請求項 4】前記符号化手段は、符号器の一連の状態を定義し、トレリス閉鎖冗長シンボルを生成するための状態マシンと呼ばれる手段を含むことを特徴とする請求項 3 記載のシステム。

【請求項 5】前記符号化レートは、 $7/8$ に等しく、前記変調は、8 次元を有する 8-D 変調であり、前記系統的重ね符号化手段は、7 個の入力シンボルを、1 個の冗長ビットを加えることによって、8 個の出力シンボルに符号化することを特徴とする請求項 3 記載のシステム。

【請求項 6】前記系統的重ね符号化手段は、2 個のシフトセル段を含み、

前記出力シンボルは、これらシフトセルから来るデータを線形結合することによって得られることを特徴とする請求項 5 記載のシステム。

【請求項 7】前記反復的復号手段は、連続して動作する少なくとも 2 つの経路を有し：第 1 の経路に沿って、前記システムは、この経路の各反復に対して、第 1 の反復的復号を：

a) 系統的重ねコードに関係するハード判定を計算するための計算手段、

b) それぞれの多次元変調サブセットと関連するハード判定の第 1 の信頼性と、系統的重ねコードのトレリスと関連する第 2 の信頼性とを計算するための計算手段、

c) それぞれの判定の前記第 1 と第 2 の信頼性のうちの最小の信頼性を選択するための選択手段、および

d) 選択された最小の信頼性とハード判定との関数として第 1 の経路の次の反復に対して使用するためのソフト判定を計算するための計算手段、を用いて遂行し；第 2 の経路に沿って、前記システムは、この経路の各反復に対して：

a) パリティコードと関係するハード判定を計算するための計算手段、

b) 第 1 の経路のハード判定の関数として第 2 の経路の各ハード判定に対する第 3 の信頼性を計算するための計算手段、および

c) 前記第 3 の信頼性、第 2 の経路の前の反復から来るソフト判定、および第 1 の経路から来るハード判定の関数としてソフト判定を計算するための手段を利用することを特徴とする請求項 3 から 6 のいずれか 1 つに記載のシステム。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】本発明は、入力データを保護するための、前記入力データに対する符号化手段および反復的復号手段を含む、デジタル伝送システムに関する。前記符号化手段は、前記反復的復号を可能にするために系統的重ねトレリス符号化(systemic convolution

al trellis coding)を遂行するための第1のサブセット、および符号化されたデータをデジタル的に変調されたシンボルに割当てするための第2のサブセットを含む。

【0002】本発明は、同様に、このようなデジタル伝送システム内において実現される入力データを保護するための方法に関する。

【0003】このシステムは、ケーブルテレビジョンあるいは衛星テレビジョンに対するデジタル伝送、地上一斉伝送、電話回線その他を通じてのデジタル伝送に利用することが可能である。

【0004】

【従来の技術】重畳符号化(convolutional coding)とデジタル変調とを組み合わせた選択的保護システムが知られている。より詳細には、このようなシステムの性能をビットエラー率の観点から向上させるためのターボコード(turbo codes)と呼ばれる新たなクラスの重畳コードが“Near Shannon limit error-correcting coding and decoding: Turbo-codes (1)”, C. BERROU, A. GLVIEUX, P. THITIMAJHIMA, Proceeding of ICC'93, Geneva, May 1993, pp. 1064-1071 において説明されている。これらターボコードは、ビットエラー率で表現した場合に、Shannonによって与えられる理論的限界に近い性能を持つ。これらは、並列に連結された重畳コードである。これら重畳コードは、系統的なパンクチャされた(punctured)帰納的コード(recursive codes)に基づく。つまり、これらは、 $1/2$ なるレートコードから導出され、これに対して入力ビットは、2個の出力ビットを与え、パンクチャリング(puncturing)は、ビット自身に適用される。このような連結は、情報を含むデータを適当にインターリーブ(interleaving)することによって実現される。

【0005】受信側では、ターボ復号と呼ばれる反復的復号が遂行される。この反復的復号は、受信されたシンボルを数回復号することで、ビットエラー率の観点からシステムの性能を向上させることから成る。この一連の反復的復号は、単一のソフト判定(soft-decision)ビタビ復号器(Viterbi decoder)によって遂行される。この復号器は、ハード判定(hard decision)を生成する従来のビタビ復号器とは異なる。上記の文献においては、復号器の出力上のソフト判定は、ハード判定信頼性比、つまり、判定が正しい尤度を生成する。この復号器に続いて出力データのデインターリーブ(deinterleaving)が行なわれる。

【0006】

【発明が解決しようとする課題】ただし、この文献において説明されるターボコードを、これに続く反復的復号と共に使用する方法では、 $1/2$ なるレートコードから来るパンクチャされた系統的コードの制約を、特に、このコードが、これらと共に用いられるデジタル変調と別個に加えられる場合は、抑制することはできない。より詳細には、これらコードは、低いスペクトル効率

(2ビット/秒/Hz以下)を持つ変調、例えば、MDP2およびMDP4位相変調にしか使用できないという短所を持つ。つまり、これらコードに対応するビットレートは、最大でも使用される帯域の2倍である。固定された占拠帯域に対してビットレートを増加させるためには、直角振幅変調(quadrature amplitude modulation: QAM)タイプの高いスペクトル効率の変調を利用することが求められる。ただし、上に説明のパンクチャされた重畳コードを、QAM変調と平置して使用する方法では、これらコードが、この変調を考慮して設計されていないために、最適な性能を達成することが不可能である。

【0007】

【課題を解決するための手段】本発明の一つの目的は、このようなデジタル伝送システムの性能を向上させること、最小のS/N比にてシステムが正しく機能することを保証すること、およびスペクトル効率を向上させることに関する。

【0008】この目的が、本発明によるシステムによって達成される。このシステムは、 $P/(Q \times M)$ なる符号化レートを持つ系統的な重畳トレリス符号化を利用する第一のサブセットを持つ。ここで、M、PおよびQは整数で、かつ、QおよびMは1より大きい。前記系統的な重畳コードが、生成コードをブロック毎に生成するために使用され、この重畳コードのトレリスは、少なくとも一つのバリティコードビットを含む冗長データの追加によって閉鎖される。前記生成コードは、系統的な重畳コードを介して符号化されたデータと結合する(データを集める)マトリックスのロウ符号化およびカラム符号化によって生成された要素によって形成される。第二のサブセットは、前記生成コードを、 2^M 個の状態を持つ次元Qを持つ多次元振幅変調と結合する。反復的復号としてブロック復号が用いられる。

【0009】生成コードおよび追加の冗長データを系統的な重畳コードに基づいて生成するために、符号器手段は、出力状態および冗長シンボルを入力状態およびデータシンボルに基づいて定義する状態マシンと呼ばれる手段から構成される。

【0010】好ましくは、この状態マシンは、系統的な重畳コードの冗長シンボルおよびトレリスを閉鎖するために使用されるシンボルを決定するための読出テーブルに結合される。

【0011】本発明によると、このシステムは、さらに、反復符号化手段を含み、これは、連続して動作する少なくとも二つの経路を持つ。そして、このシステムは、第一の経路に沿って、この経路の各反復に対して、第一の反復的復号を：

- a) 系統的な重畳コードに関係するハード判定を計算するための計算手段、
- b) 各多次元変調サブセットと関連するハード判定の第

一の信頼性、および系統的重畳コードのトレリスに関連する第二の信頼性を計算するための計算手段、

c) 各判定の前記第一と第二の信頼性の内の最小信頼性を選択するための選択手段、および

d) 前記の選択された最小信頼性とハード判定の関数として第一の経路の次の反復に対して使用するためのソフト判定を選択するための選択手段、を用いて遂行し、第二の経路に沿って、前記システムは、経路の各反復に対して、

a) パリティコードと関係するハード判定を計算するための計算手段、

b) 第一の経路のハード判定の関数として第二の経路の各ハード判定に対する第三の信頼性を計算するための計算手段、および

c) 前記第三の信頼性、第二の経路の前の反復から来るソフト判定、および第一の経路から来るハード判定の関数としてソフト判定を計算するための手段を利用する。

【0012】本発明はさらにデジタル伝送システム内で利用される入力データを保護するための方法に関する。この方法は、前記入力データに対する符号化フェーズおよび反復的復号フェーズを含む。前記符号化フェーズは、前記反復的復号を可能にするための系統的重畳トレリス符号化を遂行する第一のステップ、および第一のステップと組み合わせて使用される、前記符号化されたデータを、デジタル的に変調されたシンボルに割当てする第二のステップを含む。本発明は、前記第一のステップの際に、前記系統的重畳トレリス符号化が $P/(Q \times M)$ なる符号化レートを持つことを特徴とする。ここで、 M 、 P および Q は整数であり、かつ、 Q および M は1より大きい。前記系統的重畳コードは、生成コードをブロック毎に生成するために使用され、この重畳コードのトレリスは冗長データの追加によって閉鎖される。前記生成コードは、系統的重畳コードを介して符号化されたデータと結合する（データを集める）マトリックスのロウ符号化およびカラム符号化によって生成された要素によって形成される。前記第二のステップは、前記生成コードを、 2^M 個の状態を持つ次元 Q を持つ多次元振幅変調と結合する。前記反復的復号はブロック復号とされる。

【0013】ここでは、 $P/(Q \times M)$ なるレートの系統的重畳符号化について述べられる。このコードを、 2^M 状態（好ましくは、4 状態）の多次元デジタル変調と結合されることにより、最適な性能が達成される。この例（ $M=2$ 、 $Q=8$ 、 $P=14$ 、レート $=14/16=7/8$ ）によると、このような符号化は、ビットストリームを処理することによって得られる。より詳細には、この処理の過程において、14 個のビットあるいは7 個のシンボル（入力シンボルと呼ばれる）が符号化するために取られ、8 個のシンボル（出力シンボルと呼ばれる）が生成される。 Q 個のシンボル（それぞれ8 個の出力シンボル）が、2 個の 2^M レベルシンボル（4 レ

ベルシンボル、それぞれ $\{+1, -1, +3, -3\}$ ）を選択し、結果として、シンボル当たり M 個の出力ビット（4 レベル 4-AM の場合は、シンボル当たり 2 個の出力ビット）が生成される。この生成コードは、系統的重畳コードを、情報ビットを含むマトリックスのロウとカラムに適用することによって実現される。そして、トレリスを閉鎖するために、各ロウおよび各カラムに冗長データが追加される。最後に、マトリックスから来る 2^M -AM（例えば、4-AM）シンボルが、生成コードに従ってペアにて結合され、 2^{2M} -QAM シンボル（例えば、16-QAM シンボル）が生成される。符号化されたデータは次に従来の方法によって搬送波を用いて送信される。

【0014】受信機端におけるターボ復号の遂行を可能にするために、生成コードは、系統的であることを要求される。従って、系統的重畳コードを使用することが必要である。つまり、入力データシンボルを符号化した後に、出力上に入力シンボルが変化することなく現われることが必要とされる。

【0015】後に説明される $P/(Q \times M)$ 、例えば、 $7/8$ なるレートの重畳コードが、これらが、ある与えられた S/N 比に対して、より具体的には、4-AM あるいは 16-QAM 変調に対して、最小のエラー率を与えるように設計される。

【0016】こうして、 S/N 比と、スペクトラル効率の両方の観点から最適な性能を持つシステムが、 $7/8$ なるレートの系統的重畳符号化を最適な方法にて実現する符号器構造と多次元変調とを組み合わせることによって得られる。

【0017】本発明によると、ここに開示される系統的重畳コードを使用する生成コードは、直角振幅変調と、並置するのではなく、結合される。

【0018】このシステムは、さまざまな長所を持つ。

【0019】パンクチャ（抑止）された2進コードを M DP 4 位相変調と共に利用する従来の技術によるシステムと比較して、本発明のシステムは、2 倍高い伝送容量（2 から 4 ビット/秒/Hz の間のスペクトラル効率）を持つ。

【0020】このシステムは、生成コードの反復的復号を遂行するために単一のソフト判定復号器のみが必要とされるために、ハードウェアがあまり複雑にならない。

【0021】好ましくは、 $7/8$ なるレートの系統的重畳コードが使用され、14 個の入力ビットに対して、16 個の出力ビットが得られる。16 個の出力ビットを、入力ビット、並びに、符号器の状態（メモリ）の関数として生成するための関数が、16-QAM 変調に対して最適化される。

【0022】好ましくは $7/8$ なるレートの系統的重畳コードが選択され、これらコードの最小ユークリッド距離と 16-QAM 変調の結合の最大化を目指すことで、

10

20

30

40

50

性能の向上が図られる。これは、より具体的には、符号化と変調を結合することに関する。

【0023】本発明のこれらおよびその他の特徴が、以下に説明される実施例を参照することで、明白となるものである。

【0024】

【発明の実施の形態】図1は、本発明によるベースバンドデジタル伝送システムの略図を示す。ソース5から入来する保護されるべき入力データは、例えば、これらを、別個にアドレスされるロウ(row)とカラム(column)から形成されるメモリ10に格納することによってマトリックス(matrix)に編成される。このシステムは、ロウ符号化(coding)手段12、およびカラム符号化手段14を含む。これら2つの符号化手段によって、4-AMデジタル変調シンボル(digitally modulated symbols)に割り当てられるデータが生成される。割当て要素16は、これら4-AMシンボルを16-QAMシンボルに変換する。つまり、一対の4-AMシンボルを結合することで、16-QAMシンボルが形成される。

【0025】これらのシンボルは、従来の技術によってチャネル19を通じて受信機手段に送信される。受信されたデータは、バッファ手段20に送られ、ここで、後に説明されるループ処理が遂行される。復号器は、順番に動作する2つの経路を含む。第1の経路1は、送信されたシンボルの最下位ビット(インデックスi)についての決定に関し、第2の経路2は、送信されたシンボルの最上位ビット(インデックスj)の決定に関する。

【0026】経路1においては、バッファ手段20の出力から、データは、ビタビ復号器(Viterbi decoder)221に入る。これは、二つの出力を持ち、ハード判定出力(hard-decision output)211は、各反復に対して得られた推定シンボルおよび推定ビットiを供給し、ソフト判定出力(soft-decision output)231は、データインターバ(data interleaver)241を通過してバッファメモリ20にループバックされる。

【0027】経路2においては、バッファ手段20の出力から、データは、パリティコード復号器(parity code decoder)222に入る。これは、二つの出力を持ち、ハード判定出力212は、各反復に対して得られた推定ビット(estimated bits)jを供給し、ソフト判定出力232は、データインターバ242を通過してバッファメモリ20にループバックされる。

【0028】経路2の復号は経路1の復号の後に、経路1の復号器によって推定された受信されたシンボルとビ*

*ットiに基づいて遂行される。

【0029】推定シンボルは、反復処理が反復されるにつれて信頼性が高まる。平均的には、エラーレートとして測定されたこの改善は、経路1では4~5回の反復、経路2では2~3回の反復によって達成される。

【0030】生成コード(product code)を形成するために、このマトリックスのロウの符号化が、ロウ符号化によって、以下のように遂行される。データは、ブロックにて処理される。ここに説明される一例としての7/8なるレートの系統的コードでは、1ブロックは、 $2 \times (7N+4)$ 2個のビットを含む。メモリ10は、 $(7N+4) \times (7N+4)$ なる次元の情報マトリックス[I]を格納する。このマトリックスは、4元データシンボル(+1, -1, +3, -3)から構成され、各シンボルは、2ビットに対応する。

【0031】次に、マトリックス[I]の各ロウ(1ロウ=7N+4個のシンボル)が、系統的重畳ロウ符号器(systematic convolutional row coder)122(図3)によって符号化され、データシンボル、および冗長シンボル(redundancy symbols)の両方が生成される。(データシンボルは、符号化の前のマトリックス[I]の4元(quaternary)シンボルに対応する4-AM4-レベルシンボル(+1, -1, +3, -3)である)。各ロウに対して、7/8なるレートの重畳符号器によって生成されたN個の4-AM冗長シンボル(7N個のデータシンボルに対してN個の冗長シンボル)と、コードトレリス(code trellis)を閉鎖する(終える)ための4個の4-AM冗長シンボルが存在する。トレリスを閉鎖するためのこれら4個のシンボルは、情報マトリックス(information matrix)のロウの最後の4個のシンボルを用いて生成される。マトリックスのカラムに対しても同様の動作が遂行される。

【0032】一例として7/8なるレートの系統的重畳コード(符号器122)について説明する。7個の有益なデータシンボルから、7個の有益なシンボル(系統的重畳コード)と、これに追加しての1個の冗長シンボルから構成される8個のシンボルが生成される。N+1ブロック(1ブロックは、7個のデータシンボル、つまり、14ビットを持つ)によって形成されるマトリックスの第1のロウについて考える。これらは、以下の通りである：

【0033】

【数1】

$$(I_{1,1}, \dots, I_{1,7}), (I_{1,8}, \dots, I_{1,14}), \dots, \\ (I_{1,7N-6}, \dots, I_{1,7N}), (I_{1,7N+1}, \dots, I_{1,7N+4}).$$

あるシンボル、例えば、 $I_{1,1}$ は、二つのビット $j_{1,1}$ および $i_{1,1}$ から構成される。

【0034】図3の略図に従って実現される生成コードは、表1に示すようなマトリックスを与える。

【0035】

* * 【表1】

$I_{1,1} \dots I_{1,7N+4}$ \dots $I_{m,1} \dots I_{m,7N+4}$ \dots $I_{7N+4,1} \dots I_{7N+4,7N+4}$	$R_{1,1} \dots R_{1,N}$ \dots $R_{m,1} \dots R_{m,N}$ \dots $R_{7N+4,1} \dots R_{7N+4,N}$	$TR_{1,1} \dots TR_{1,4}$ \dots $TR_{m,1} \dots TR_{m,4}$ \dots $TR_{7N+4,1} \dots TR_{7N+4,4}$
$C_{1,1} \dots C_{1,7N+4}$ \dots $C_{m,1} \dots C_{m,7N+4}$ \dots $C_{N,1} \dots C_{N,7N+4}$		
$TC_{1,1} \dots TC_{1,7N+4}$ \dots $TC_{4,1} \dots TC_{4,7N+4}$		

(表1)

各ブロックに対して、系統的重畳コードの冗長シンボル (つまり、2ビット) が計算される。ブロック1は $R_{1,1}$ を生成し、ブロック2は $R_{1,2}$ を生成し、そしてこれが、ブロックNによって、 $R_{1,N}$ が生成されるまで行なわれる。

【0036】符号器の初期状態は、0であるものと想定する。 $R_{1,N}$ を生成した後に、符号器122は、状態 $\sigma(N)$ となる。トレリス閉鎖手段128を用いてトレリスが閉鎖される。これは、8-状態重畳コード(8-state convolutional code)のトレリスにおける遷移に対応する4個のシンボル $TR_{1,1}$, $TR_{1,2}$, $TR_{1,3}$, $TR_{1,4}$ を生成することで、トレリスの状態を0に強制することによって行なわれる。そして、この遷移の後、状態 $\sigma=0$ に到達する。上記は、8-状態コードの場合についてであるが、より多くの状態を持つコードに対しては、より多くのトレリス閉鎖シンボル(trellis closing symbols)を使用することが必要となる。シンボル $TR_{1,1}$, $TR_{1,2}$, $TR_{1,3}$, $TR_{1,4}$ は、4個のデータシンボル($I_{1,7+1}, \dots, I_{1,7N+4}$)とロウが符号化された後のトレリス $\sigma(N)$ の状態との関数であり、後に説明されるように、 $\sigma(N)$ にてアドレスされるテーブルによって

30 生成される。

【0037】符号化の後に、表1に示されるような4-AMシンボルのマトリックスが得られる。ここで： $I_{m,n}$ は、4-AMデータシンボルから構成される $(7N+4) \times (7N+4)$ なる次元を持つマトリックスであり； $R_{m,n}$ は、7/8コードによって生成されたロウ冗長シンボルに対応する4-AMシンボルから構成される $(7N+4) \times N$ なる次元(dimension)のマトリックスであり； $TR_{m,n}$ は、4-AMトレリス閉鎖ロウ冗長シンボル(trellis closing row redundancy symbols)から構成される $(7N+4) \times 4$ なる次元のマトリックスであり； $C_{m,n}$ は、カラム冗長シンボルから構成される $(N \times 7N+4)$ なる次元のマトリックスであり； $TC_{m,n}$ は、トレリス閉鎖カラム冗長シンボルから構成される $(4 \times 7N+4)$ なる次元を持つマトリックスである。

【0038】ロウ符号器12は、以下のように動作する(図3)。入力シンボル $I_{m,n}$ は、バッファ120に入り、次に、7/8なるレート(率)の符号器122に入る。符号器122は、7個のデータシンボルから成る1ブロックに対して、1個の冗長シンボルを生成する。マルチプ

50

レクサ124は、出力上のシンボルをシリアル化する。こうして、マルチプレクサ124は、出力131上に、順番に、8個のシンボルから成るグループを供給する。ここで、各グループは、7個のデータシンボルと、そのブロックに対応する1個の冗長シンボルから構成される。次に、マルチプレクサは、8個のシンボルから成る全てのグループを順番に供給する。全てのグループが抽出された後に、ロウの最後の4個のデータシンボルが抽出される。その後、トレリス閉鎖要素128によって生成された4個のトレリス閉鎖シンボルが抽出されるが、これによって、シンボルの閉鎖(終わり)が決定される。コントローラ17は、これらの動作を制御する。

【0039】他のロウに対しても同様のプロセスが遂行される。カラムに対しても類似するプロセスが実現される(カラム符号器14)。第1のカラムの第1のブロックは、シンボル: $I_{1,1}, I_{2,1}, \dots, I_{7N+4,1}$ から形成される。7個のシンボルから成るブロックのそれぞれに、1個の冗長シンボルが対応する。例えば、冗長シンボル $C_{1,1}$ は、第1のブロックに対応し、冗長シンボル $C_{N,1}$ は、最後のブロックに対応する。最初のカラムに
 20 対するトレリス閉鎖冗長シンボルは、シンボル $T_{C1,1}, T_{C2,1}, T_{C3,1}, T_{C4,1}$ であるが、これらは、シンボル $I_{7N+1,1}, \dots, I_{7N+4,1}$ に基づいて生成される。初期データシンボル、ロウ符号化冗長シンボル、カラム符号化冗長シンボル、トレリス閉鎖シンボルから成る全てのデータによって生成コードが形成される。

【0040】この生成コードの1つのロウによって、シンボルの1つのブロックが形成され、この一つのブロックが、4-AM変調シンボルを符号化するために用いられる。初期データシンボルは、生成コードにおいて、これらがカラム毎に読まれてもロウ毎に読まれても変わらないために、生成コードのカラムの利用に当たっては、初期データシンボルを再度使用する必要はない。このために、生成コードのカラムについては、冗長シンボルのみが、4-AM変調シンボルを符号化するために利用される(カラム符号器14)。生成コードのロウおよびカラムから来る4-AMシンボルが、要素16内で、16-QAM変調を符号化するために用いられる。

【0041】システムの性能を向上させるために、こう

して計算されたばかりの冗長シンボルについても符号化することも可能である。

【0042】例えば、マトリックスのカラム $R_{m,n}$ および $TR_{m,n}$ を符号化することで(ロウ冗長カラム符号化)、それぞれ、 $(N \times N)$, $(N \times 4)$, $(4 \times N)$, (4×4) なる次元を持つマトリックス $R_{Cm,n}, TR_{Cm,n}, RCT_{m,n}, TRCT_{m,n}$ が生成される(テーブル=)。マトリックスのロウ $C_{m,n}$ および $TC_{m,n}$ についても同様に符号化され(カラム冗長ロウ符号化)、それぞれ、 $(N \times N)$, $(N \times 4)$, $(4 \times N)$, (4×4) なる次元を持つマトリックス $C_{Rm,n}, CRT_{m,n}, TCR_{m,n}, TCRT_{m,n}$ が生成される。

【0043】この場合、上に説明したのと同じの要領で、 $(7N+4) \times (7N+4)$ なる次元を持つ情報マトリックスが、ロウおよびカラムに従って符号化され、ロウ冗長(シンボル $R_{m,n}, TR_{m,n}$)、およびカラム冗長(シンボル $C_{m,n}, TC_{m,n}$) が得られる。次に、システムの性能を向上させるために、カラムに対してロウ冗長が符号化され、ロウに対してカラム冗長が符号化される。例えば、 $7N+4$ 個のシンボル ($R_{1,1}, \dots, R_{C7N+4,1}$) から成るブロックを符号化することによって、 $7/8$ なるレートのコード冗長から来る第1のN個のカラム冗長シンボル ($RC_{1,1}, \dots, RC_{N,1}$) と、トレリスを閉鎖するための4個の冗長シンボル ($RCT_{1,1}$ から $RCT_{4,1}$) とが生成される。ロウ冗長の他のカラム ($R_{1,n}, \dots, R_{7N+4,n}$) に対しても、 $n=1, \dots, N$ について、および、($TR_{1,1}, \dots, TR_{7N+4,1}$) から ($TR_{1,4}, \dots, TR_{7N+4,4}$) までの4個のカラムについて、同様に符号化される。カラム冗長のロウ、つまり
 30 ($C_{m,1}, \dots, C_{m,7N+4}$) なるロウについても、 $m=1, \dots, N$ について、および、($TC_{1,1}, \dots, TC_{1,7N+4}$) から ($TC_{4,1}, \dots, TC_{4,7N+4}$) までの4個のロウについて、同様に符号化される。

【0044】こうして、以下のマトリックスが得られる:

【0045】

【表2】

$I_{1,1} \dots I_{1,n} \dots I_{1,7N+4}$ $I_{n,1} \dots I_{n,n} \dots I_{n,7N+4}$ $I_{7N+4,1} \dots I_{7N+4,n} \dots I_{7N+4,7N+4}$	$R_{1,1} \dots R_{1,N}$ $R_{n,1} \dots R_{n,N}$ $R_{7N+4,1} \dots R_{7N+4,N}$	$TR_{1,1} \dots TR_{1,4}$ $TR_{n,1} \dots TR_{n,4}$ $TR_{7N+4,1} \dots TR_{7N+4,4}$
$C_{1,1} \dots C_{1,n} \dots C_{1,7N+4}$ $C_{n,1} \dots C_{n,n} \dots C_{n,7N+4}$	$RC_{1,1} \dots RC_{1,N}$ $RC_{n,1} \dots RC_{n,N}$	$TRC_{1,1} \dots TRC_{1,4}$ $TRC_{n,1} \dots TRC_{n,4}$
$TC_{1,1} \dots TC_{1,n} \dots TC_{1,7N+4}$ $TC_{4,1} \dots TC_{4,n} \dots TC_{4,7N+4}$	$RCT_{1,1} \dots RCT_{1,N}$ $RCT_{4,1} \dots RCT_{4,N}$	$TRCT_{1,1} \dots TRCT_{1,4}$ $TRCT_{4,1} \dots TRCT_{4,4}$
	$CR_{1,1} \dots CR_{1,N}$ $CR_{n,1} \dots CR_{n,N}$	$CRT_{1,1} \dots CRT_{1,4}$ $CRT_{n,1} \dots CRT_{n,4}$
	$TCT_{1,1} \dots TCT_{1,N}$ $TCT_{4,1} \dots TCT_{4,N}$	$TCRT_{1,1} \dots TCRT_{1,4}$ $TCRT_{4,1} \dots TCRT_{4,4}$

(表2)

この表において：

- マトリックス [I] は、入力データシンボルを含み、
- マトリックス [R] は、マトリックス [I] のロウ冗長マトリックスであり、
- マトリックス [C] は、マトリックス [I] のカラム冗長マトリックスであり、
- マトリックス [TR] は、マトリックス [I] のロウに対するトレリス閉鎖冗長マトリックス(trellis closing redundancy matrix)であり、
- マトリックス [TC] は、マトリックス [I] のカラムに対するトレリス閉鎖冗長マトリックスであり、
- マトリックス [RC] は、マトリックス [R] のカラム冗長マトリックスであり、
- マトリックス [CR] は、マトリックス [C] のロウ冗長マトリックスであり、
- マトリックス [TCR] は、マトリックス [TC] のロウ冗長マトリックスであり、
- マトリックス [RCT] は、マトリックス [R] のカラムに対するトレリス閉鎖冗長マトリックスであり、
- マトリックス [TRC] は、マトリックス [TR] のカラム冗長マトリックスであり、
- マトリックス [TRCT] は、マトリックス [TR] のカラムに対するトレリス閉鎖マトリックスである。

【0046】—マトリックス [CRT] は、マトリックス [C] のロウのトレリス閉鎖マトリックスであり、

—マトリックス [TCRT] は、マトリックス [TC] のロウ符号化のトレリス閉鎖マトリックスである。

【0047】次に、マトリックスのロウを符号化するための図3に示されるロウ符号器12の動作の詳細についてより詳細に説明する。

【0048】バッファ120内に、符号化すべきロウが格納される。一つのロウは、 $(14N+8)$ 個のビット、つまり、 $(7N+4)$ 個の4元データシンボルから構成される。例えば、 k 番目のブロックに対しては、このバッファは、14個のビット（つまり、7個のシンボル）を持つ N 個のブロック $(i_1(k), j_1(k), \dots, i_7(k), j_7(k))$ と、8個のビット（つまり、4個のシンボル）から成る1つのブロックを格納する：

【0049】

【数2】

$$\begin{aligned}
 &i_1(N+1), j_1(N+1) \\
 &i_2(N+1), j_2(N+1) \\
 &i_3(N+1), j_3(N+1) \\
 &i_5(N+1), j_5(N+1)
 \end{aligned}$$

14個のビットから成る N 個のブロックには、次々と、 $7/8$ なるレートの系統的重畳符号化が施される（符号器122）。各ブロック（ k 番目のブロック）に対し

て、符号器122は、図8に示される割当てを実現することによって、 $k=1, \dots, N$ に対して、そのブロックの14個の情報ビットと、一つの冗長シンボル $U_8(k)$ に対応する、7個の4-AMシンボル $U_1(k) \sim U_7(k)$ を生成する。

【0050】14個の情報ビットから成る各ブロックは、 $7/8$ なるレートの符号トレリスにおける遷移に対応し、 $N+1$ 番目のオーダの最後のブロックは、このトレリスを閉鎖する遷移に対応する。

【0051】8個の情報ビット $\{i_1(N+1), j_1(N+1), i_2(N+1), j_2(N+1), i_3(N+1), j_3(N+1), i_5(N+1), j_5(N+1)\}$ 、つまり、ロウの8個の最後のビットである4個の4元シンボルに基づいて、図8によって与えられる割当てられるとちょうど同様な、4-AMシンボル、つまり、 $U_1(N+1), U_2(N+1), U_3(N+1), U_5(N+1)$ が生成され、次に、トレリスの閉鎖するための4個の冗長シンボル $U_4(N+1), U_6(N+1), U_7(N+1), U_8(N+1)$ が計算される。これら冗長シンボルは、 N 番目のブロックの符号化の後、符号器の状態 n_{N+1} と8個のビット $\{i_1(N+1), j_1(N+1), i_2(N+1), j_2(N+1), i_3(N+1), j_3(N+1), i_5(N+1), j_5(N+1)\}$ の両方の関数である。これらは、トレリス閉鎖要素128によって決定される。

【0052】カラム符号器は、 $7/8$ なるレートの系統的重畳符号化の出力に冗長シンボルのみが存在する点を除いてロウ符号器と同一である。

【0053】図4は、冗長の二重符号化を可能にする符号器の略図を示す。図1と同一の要素は、同一の番号で示す。説明の例においては、メモリM10は、 $(7N+4)^2$ 個のシンボルを含む。もう一つのメモリMRは、 $(7N+4)(N+4)$ 個のロウ冗長シンボル(シンボル $R_{m,n}, TR_{m,n}$)を含む。もう一つのメモリMCは、 $(N+4)(7N+4)$ 個のカラム冗長シンボル(シンボル $C_{m,n}, TC_{m,n}$)を含む。これらメモリMRおよびMCは、メモリ10の内容のロウおよびカラム符号化を終えた後にロードされる。次に、カラムに従ってMRの内容の符号化が遂行され、ロウに従ってMCの内容の符号化が行われ、それぞれ、 $(N+4)^2$ 個の4-AMシンボル($RC_{m,n}, TRC_{m,n}, RCT_{m,n}, TRCT_{m,n}$)、および $(N+4)^2$ 個の4-AMシンボル($CR_{m,n}, CRT_{m,n}, TCR_{m,n}, TCRT_{m,n}$)が生成される。スイッチ13およびスイッチ15は、コントローラ17の制御下で動作を遂行することを確保する。

【0054】最後に、表1あるいは表2の各生成コードマトリックスに対して形成された全ての4-AMシンボルがペアにて結合され、これらがチャネルを通じて16-QAM群(constellation)の複合シンボルの形式にて(整列順序にて)伝送される。

【0055】冗長符号化しない場合の生成コードのレートは：

$$\rho_1 = (7N+4)^2 / \{(7N+4)^2 + 2(7N^2 + 32N + 16)\}$$

となる。 $N=20$ の場合は、レートは、0.765となり、スペクトラル効率はこのレートの4倍、つまり、3.06ビット/秒/Hzとなる。

【0056】二重冗長符号化を行なった場合は、生成コードのレートは：

$$\rho_2 = (7N+4)^2 / \{16(N+4)(N+1) + (7N+4)^2\}$$

となる。

【0057】この場合は、 $N=20$ の場合、レートは、0.72となり、スペクトラル効率は、2.88ビット/秒/Hzとなる。冗長を符号化した場合は、0.18ビット/秒/Hzのスペクトラル効率の損失が生ずるが、ただし、システムの性能は向上する。

【0058】図8は、ビット i, j の4-AM群(constellation)のシンボルへのビット割当てを示す。

【0059】一次元の4-AM群(constellation)は、セット(set) $A_0 = \{-3, -1, +1, +3\}$ によって表される。セット A_0 の第1の分割レベルは、2つのサブセット(sub-set) B_0 および B_1 から構成される。サブセット $B_0 = \{3, -1\}$ とされ、サブセット $B_1 = \{1, -3\}$ とされる。あるシンボル、例えば、 $I_{1,1}$ は、2個のビット $i_{1,1}$ 、および $j_{1,1}$ から構成される。この第1のレベルには、ビット i が割当てられ、 B_0 に対しては $i=0$ とされ、 B_1 に対しては $i=1$ とされる。セット A_0 の第2の分割レベルは、4つのサブセット C_0, C_1, C_2, C_3 から構成される。サブセット $C_0 = \{+3\}$ 、 $C_1 = \{+1\}$ 、 $C_2 = \{-1\}$ 、 $C_3 = \{-3\}$ とされる。この第2のレベルにインデックス j が割当てられ、 C_0 あるいは C_1 に対しては $j=0$ とされ、 C_2 あるいは C_3 に対しては、 $j=1$ とされる。セット C のインデックスは、それらを構成する分割に割当てられたインデックス j, i の10進値、つまり、 $2j+i$ とされる。

【0060】8-D群は、おのおのが A_0 に属する8個のシンボルから成るブロックのセットであるものと定義される。図5は、 A_0 が1次元(1-D)4-AM群である8-D群(A_0)8の分割ツリー(partition tree)を表す。

【0061】セット(A_0)⁸は、(二乗)最小ユークリッド距離に対する d_{02} を持つ。このセットが、 $2d_{02}$ の最小距離を持つ2個のサブセットに分割される。各サブセットに、0あるいは1の値を持つビット Y_0 が割当てられる。

【0062】

【数3】

$$\sum_{k=1}^8 i_k = 0 \pmod{2};$$

となるような B_{i1}, \dots, B_{i8} に属するすべてのブロックによって形成されるサブセットに対しては、 $Y_0=0$ が割当てられ;

【0063】

【数4】

$$\sum_{k=1}^8 i_k = 1 \pmod{2}$$

となるような $B_{i1} \dots B_{i8}$ に属するすべてのブロックによって形成されるサブセットに対しては、 $Y_0=1$ が割当てられる。

【0064】 こうして、 $Y_0=0$ は、1-Dサブセット内に偶数個の要素を含む全ての4-AMシンボルのブロック ($U_1 \dots U_8$) のセット、つまり、 $B_1 = \{+1, -3\}$ に対応する。

【0065】 同様に、 $Y_0=1$ は、1-Dサブセット内に奇数個の要素を含む全ての4-AMシンボルブロック ($U_1 \dots U_8$) のセット (集合)、つまり、 $B_0 = \{-1, +3\}$ に対応する。

【0066】 次に、それぞれの8-Dサブセットが、2個の別の8-Dサブセットに分割され、これが反復される。

【0067】 各分割レベルに以下のように1ビットが割当てられる:

- Y_0 が第1のレベルに割当てられ;
- Y_1 が第2のレベルに割当てられ;
- Y_2 が第3のレベルに割当てられ;
- Y_3 が第4のレベルに割当てられる。

【0068】 以下では、 $Y_0=0$ に対応するサブセットについてのみ説明される。

【0069】 4番目の分割レベルの後に、8個のサブセット: $S_0, S_2, S_4, S_6, S_8, S_{10}, S_{12}, S_{14}$ (S_i, i は偶数) が得られる。

【0070】 サブセット S_i のインデックス i は、 S_i に割当てられた4ビット Y_3, Y_2, Y_1, Y_0 の10進値とされる。

【0071】 各サブセットは、形式 $U(B_{i1} \dots B_{i8})$ (サブセットの和) の形式を持ち、インデックス (i_1, \dots, i_8) のパリティ関係によって定義される。各サブセット S_i における最小 (二乗) 距離は、 $4-d_0^2$

である。各サブセットは、8個の4-AMシンボルから成る $2^{12}=4096$ 個のブロックを含む。

【0072】 各サブセット S_i は、16個のサブセット ($B_{i1} \dots B_{i8}$) から構成され、各サブセット ($B_{i1} \dots B_{i8}$) は、 $2^8=256$ 個の8-Dブロックを含む。

【0073】 最後に、各サブセット S_i が、2個のサブセット $S_{i,0}$ と、 $S_{i,1}$ とに分割される。ここで:

- $S_{i,0}$ は、そのブロックの最後の4-AMシンボルが正である (すなわち C_0 あるいは C_1 に属する) ような S_i のブロックのサブセットであり、

- $S_{i,1}$ は、そのブロックの最後の4-AMシンボルが負である (すなわち C_2 あるいは C_3 に属する) ような S_i のブロックのサブセットである。

【0074】 こうして、 $S_{i,0}$ は、 S_i の内の、最後のシンボル U_8 が C_0 あるいは C_1 となる、つまり、最後のシンボルが、ビット $j_8=0$ に対応する8-Dブロックのセットに対応する。 $S_{i,0}$ は、 2^{11} 個の要素を含む。

【0075】 $S_{i,1}$ は、 S_i の内の、 $j_8=1$ となるような集合 (U_1, \dots, U_8) から成る。

【0076】 例えば、サブセット $S_{6,1}$ は、 (A_0) 8の内の、 $j_8=1$ と、ビット $i_1 \sim i_8$ の以下の4つのパリティ関係を有する (verify) ブロック (U_1, U_2, \dots, U_8) のセットによって定義される:

【0077】

【数5】

$$i_1 \oplus i_3 \oplus i_5 \oplus i_7 = 0,$$

$$i_1 \oplus i_2 \oplus i_5 \oplus i_6 = 1,$$

$$i_1 \oplus i_2 \oplus i_3 \oplus i_4 = 1.$$

$$\sum_{p=1}^8 i_p = 0$$

ここで、 (i_p, j_p) は、4-AMシンボル U_p に割当てられたビットである (図8)。

【0078】 一般的なケースでは、 i が偶数である場合は、 $S_{i,j}$ は、4-AMシンボルの内の、以下のパリティ関係を有する (verify) ブロック $U = (U_1, \dots, U_8)$ のセット (集合) である:

【0079】

【数6】

$$\begin{aligned}
 i_1 \oplus i_3 \oplus i_5 \oplus i_7 &= Y_3 \\
 i_1 \oplus i_2 \oplus i_5 \oplus i_6 &= Y_2 \\
 i_1 \oplus i_2 \oplus i_3 \oplus i_4 &= Y_1 \\
 i_1 \oplus i_2 \oplus i_3 \oplus i_4 \oplus i_5 \oplus i_6 \oplus i_7 \oplus i_8 &= Y_0 = 0 \\
 j_8 &= j
 \end{aligned}$$

ここで、 $(Y_3 Y_2 Y_1 Y_0)$ は、 i のビット表現である。すなわち： $i = 8 Y_3 + 4 Y_2 + 2 Y_1 + Y_0$ である。ここで、 $Y_0 = 0$ である。また、 (i_p, j_p) は、ブロック U_p の p 番目のシンボルのビット割り当て (bit allocation) であり、

$U_p = 3$ または -1 であれば、 $i_p = 0$
 $U_p = 1$ または -3 であれば、 $i_p = 1$
 $U_p = 1$ または 3 であれば、 $j_p = 0$
 $U_p = -1$ または -3 であれば、 $j_p = 1$
 である。

【0080】 こうして、全ての $S_{i,j}$ ブロック (2¹¹個のブロック) を生成するためには、11個のビット $i_1, i_2, i_3, i_5, j_1, j_2, j_3, j_4, j_5, j_6, j_7$ の全ての可能な組合せを走査することで十分である。そして、任意の組合せに対して、他のビット (i_4, i_6, i_7, i_8, j_8) は、上に定義されたサブセット $S_{i,j}$ のパリティ関係に基づいて計算することが可能である。

【0081】 本発明によると、7/8なるレートの系統的重畳符号 (systematic convolutional code) は、以下のようにして形成される：サブセット $S_{i,j}$ が、トレリスの並列のブランチに割当てられる。また、偶数の i を持つサブセット $S_{i,j}$ が利用される。これは、最小 (二乗) 距離が、 $4 d_o^2$ に等しくなることを保証する。ここで、 d_o は、4-AM群の最小距離である。

【0082】 7/8なるレートを持つコードの場合は、符号器のある与えられた状態から来る、8個の送信された4-AMシンボル (つまり、16ビット) に対する14個の入力ビット (この内の3個は符号化されたビットで、11個は符号化されていないビット) に対応する、8個の別個の遷移が存在することが必要である。これによって、8-状態コードのトレリスが完全に接続される。

【0083】 系統的コードとするためには、つまり、8-Dブロックの内の送信された最初の7個の4-AMシンボルが、7個のデータシンボル (あるいは14個のビット) となるようにするためには、2¹⁴個の遷移が、ある状態に基づいて、8個の4-AMシンボルの2¹⁴個のブロックを生成し、最初の7個が、2¹⁴個の可能な組合せを採用するようすることが必要である。このためには、一つの状態から来る8個の別個のブランチに対して、8個のサブセット $S_{i,j}$ を割当ててことで十分である。ここで、 $i = 0, 2, 4, 6, 8, 10, 12, 14$

4とされ、 j は、任意の値とされる。

【0084】 図6は、7/8なるレートの最適な系統的重畳コードのトレリスを示す。図において、 σ_k は、瞬間 k における符号器の状態であり、 σ_{k+1} は、瞬間 $k+1$ における符号器の状態である。状態は、符号器のメモリ内の3個のビット (3個のシフトレジスタ) によって定義される。この図は、さらに、サブセット $S_{i,j}$ のさまざまなトレリス遷移への割当てを示す。このトレリスが、以下のように分析される：符号器は、瞬間 k における状態 $\sigma(k) : \sigma_k = (\sigma_k^2, \sigma_k^1, \sigma_k^0)$ に基づいて、瞬間 $(k+1)T$ における別の状態 $\sigma(k+1) : \sigma_{k+1} = (\sigma_{k+1}^2, \sigma_{k+1}^1, \sigma_{k+1}^0)$ へと進む。

【0085】 こうして、符号器は、状態 [111] から、8個の状態 [000] [001] [010] [011] [100] [101] [110] [111] のいずれか1つに進む。

【0086】 例えば、状態001から来る第3の遷移、つまり、遷移001→010は、 $S_{10,1}$ に対応することが示される。同様に、トレリスの全体が分析される。

【0087】 図7は、7/8なるレートの符号器122 (図3) のブロック図を示す。符号器の入力上の14個のビット、つまり、 $(i_1(k), j_1(k))$ ($1 = 1, \dots, 7$) は、瞬間 k において、図7に示すように、送信されたブロック $U_1(k), \dots, U_7(k)$ の最初の7個のシンボルを選択する。冗長シンボル $U_8(k)$ 、つまり、 $(i_8(k), j_8(k))$ を計算するために、最初に、3個のビット $Y_3(k), Y_2(k), Y_1(k)$ に同じように対応するサブセット S_i のインデックス i (偶数の i) が計算される。ここで、 $i = 8 Y_3 + 4 Y_2 + 2 Y_1 + Y_0$ 、そして、 $Y_0 = 0$ である。 $i_8(k)$ についても計算されるが、これは、ビット $i_1(k), \dots, i_7(k)$ のパリティビットである。ビット $j_8(k)$ は、その入力に対してビット Y_1, Y_2, Y_3 を持つ、3/4なるレートの系統的重畳符号器230の冗長ビットである。ビット (i_8, j_8) によってシンボル $U_8(k)$ が選択される。この3/4なるレートの符号器の出力においては、符号器の次の状態 σ_{k+1} が得られるが、これは、ロウあるいはカラムの終端の所でトレリスを閉鎖するために用いられる ($k = N$)。

【0088】 ビット $Y_1(k), Y_2(k), Y_3(k)$

および $i_8(k)$ は、ビット $i_1(k)$, $i_2(k)$, \dots , $i_7(k)$ の関数として、以下の式に従って計算される：

*【0089】
【数7】

$$\begin{aligned} Y_1(k) &= i_1(k) \oplus i_2(k) \oplus i_3(k) \oplus i_4(k) \\ Y_2(k) &= i_1(k) \oplus i_2(k) \oplus i_5(k) \oplus i_6(k) \\ Y_3(k) &= i_1(k) \oplus i_3(k) \oplus i_5(k) \oplus i_7(k) \\ i_8(k) &= i_1(k) \oplus i_2(k) \oplus i_3(k) \oplus i_4(k) \oplus i_5(k) \oplus i_6(k) \oplus i_7(k) \end{aligned}$$

これらの式は、サブセット S_i のビット Y_0 , Y_1 , Y_2 , Y_3 へのビット割当て (図5) から導出され、これが、要素210内で実現される。

【0090】符号器のこの構造 (図7) は、 $S_{i,j8}$ ($i = 8Y_3 + 4Y_2 + 2Y_1$) の全ての8-Dブロックを、トレリスの状態 σ から別の状態への並列遷移に割当ててることを可能にする。

【0091】図7に示した重畳符号器は、テーブル (図11) を介して実現することも、シフトレジスタ (図9と図10) を介して実現することも、あるいはコンピュータのソフトウェアを介して実現することも可能である。

【0092】図11に示すように、テーブル220 (メモリ) は、冗長ビット $j_8(k)$ および 将来の状態 σ_{k+1} (2^v 個の状態を持つコードの場合は v 個のビット) を、 σ_k , $Y_1(k)$, $Y_2(k)$, $Y_3(k)$ の関数として与える。テーブル220は、 Y_1 , Y_2 , Y_3 と共に自身の出力 σ_{k+1} も受信する。後者は、遅延セル222を通じて入力のループバックされる。

【0093】図9および図10は、それぞれ、最適な8-状態符号器 ($v=3$)、および16-状態符号器 ($v=4$) を示す。8-状態符号器の場合は、加算器セル3121, 遅延セル3101, 加算器セル3122, 遅延セル3102, 加算器セル3123, 遅延セル3103, 加算器セル3124により連鎖が形成され、このシステムの出力は、 $j_8(k)$ を生成する。

【0094】加算器セル3121は、 $Y_1(k)$ を受信し、加算器セル3122は、 $Y_3(k)$ を受信し、加算器セル3123は、 $Y_2(k)$, $Y_3(k)$ を受信し、加算器セル3124は、 $Y_1(k)$, $Y_2(k)$, $Y_3(k)$ を受信する。

【0095】16-状態符号器 (図10) の場合は、システムは、8-状態システムの場合と同一の要素によって形成されるが、これに、遅延セル3104、および加算器セル3125が追加される。

【0096】図12は、トレリス閉鎖構成を示す。符号器の状態 σ を考え、サブセット $S_{i,j8}$ (8-Dサブセット) が、トレリスを閉鎖する遷移 (2^{11} 個の並列ブランチ) に割当てられるものとする。ここで、 $i = 8Y_3 + 4Y_2 + 2Y_1$ である。

【0097】トレリスが閉鎖され得る 2^{11} 個のブランチが存在するために、ブランチを、4個のデータシンボル U_1 , U_2 , U_3 , U_5 、すなわち、8個の情報ビット i_1 , j_1 , i_2 , j_2 , i_3 , j_3 , i_5 , j_5 の関数として探すことができる。

【0098】トレリスの閉鎖遷移 (closing transition) の際に、それぞれ、ビット (i_1 , j_1), (i_2 , j_2), (i_3 , j_3), (i_5 , j_5) に対応する4個のデータシンボル U_1 , U_2 , U_3 , U_5 が送信される。

【0099】計算は、2つのステップ (図12) で行なわれる：ロウあるいはカラムの符号化の終端における符号器の状態 σ_{N+1} に基づいて、ビット $Y_1(N+1)$, $Y_2(N+1)$, $Y_3(N+1)$, $j_8(N+1)$ が、表3を含むテーブル210に基づいて生成される。

【0100】

【表3】

σ (2進数) ($\sigma^2, \sigma^1, \sigma^0$)	σ (10進数)	$S_{i,j8}$	Y_3, Y_2, Y_1	j_8
0 0 0	0	$S_{0,0}$	0 0 0	0
0 0 1	1	$S_{0,1}$	0 1 1	1
0 1 0	2	$S_{2,1}$	0 0 1	1
0 1 1	3	$S_{4,1}$	0 1 0	0
1 0 0	4	$S_{12,0}$	1 1 0	0
1 0 1	5	$S_{10,1}$	1 0 1	1
1 1 0	6	$S_{14,1}$	1 1 1	1
1 1 1	7	$S_{6,0}$	1 0 0	0

(表3)

ビット $Y_1(N+1)$, $Y_2(N+1)$, $Y_3(N+1)$, $j_8(N+1)$ に基づいて、ビット (i_4 , j_4), (i_6 , j_6), (i_7 , j_7), (i_8 , j_8) ($N+1$) が生成され、これから計算手段240において、シンボル $U_4(N+1)$, $U_6(N+1)$, $U_7(N+1)$, $U_8(N+1)$ が生成される。これらビットは、以下のように計算される：

【0101】

【数8】

$$\begin{aligned}
 i_4(N+1) &= i_1(N+1) \oplus i_2(N+1) \oplus i_3(N+1) \oplus Y_1(N+1) \\
 i_6(N+1) &= i_1(N+1) \oplus i_2(N+1) \oplus i_5(N+1) \oplus Y_2(N+1) \\
 i_7(N+1) &= i_1(N+1) \oplus i_3(N+1) \oplus i_5(N+1) \oplus Y_3(N+1) \\
 i_8(N+1) &= i_2(N+1) \oplus i_3(N+1) \oplus i_5(N+1) \oplus Y_1(N+1) \oplus Y_2(N+1) \oplus Y_3(N+1)
 \end{aligned}$$

ビット j_4 、および j_6 は、どのような値を取ることもでき、これらは、例えば、0 ($j_4 = j_6 = 0$) と決められる。

【0102】ビット j_7 は、先行する遷移の全てのビット * 10 【数9】

$$j_7(N+1) = \sum_{k=1}^N \sum_{l=1}^7 j_l(k) \oplus \sum_{l=1}^6 j_l(N+1)$$

これは、図12に示す計算手段によって実現される。好ましくは、これら計算は、コンピュータのソフトウェアを介して実現される。

【0104】復号手段が、図1に示される。復号は、二つの異なる段内で、複数のステップにて遂行される。第一の段においては、ビット i 、つまり、送信されたシンボルの最下位ビット (LSB) の反復的復号が遂行される。

【0105】チャネルを通じて受信されたシンボルのマトリックスが、メモリ20に格納される。このマトリックスは、送信されたシンボルのマトリックスに対応する (マトリックスの次元は、送信機端で符号化されたマトリックスの次元と同一である)。

【0106】ロウに続いて、カラムが、順番に、独立して、復号される。各反復において、ビット i (ビットマトリックス) に関する判定 (接続21) が生成され、次の反復の際に使用される実数シンボルマトリックス (判定の信頼性) がメモリ20内に格納される (接続23)。デインタリーバ24によってデータが遅延される。

【0107】一ロウおよびカラムの復号のために、この好ましい例においては、7/8なるレート of の8-D重畳コードに対応するソフト判定ビタビ復号器22が使用される。

【0108】第2の段においては、第1のステップをある回数 (通常は、3あるいは4回) だけ反復した後に、ステップ2 (接続25) に進み、ビット j 、つまり、送信されたシンボルの最上位ビット (MSB) が復号される。

【0109】次に、ロウに続いて、カラムが、メモリ20内に格納された実数マトリックス、および第一の段において検出されたビット i のマトリックスが用いられるという意味で反復的に復号される。

【0110】ロウあるいはカラムの復号器22として、生成コードの各ロウおよびカラムに対して7/8なるレート of の重畳コードのトレリスを閉鎖するとき用いられたパリティコードのソフト判定復号器が用いられ

* ト (j_1, \dots, j_7) のパリティビットであり、以下によって与えられる:

【0103】

【数9】

る。

【0111】最初に、第1の段 (経路1) の動作について説明する。図13は、送信された4-AMシンボルのLSBビット (ビット i) に対する二重判定ビタビ復号手段22の一般回路図を示す。これは:

- マトリックスを計算するためのサブセット220
- ハード判定を生成するビタビ復号器222
- 判定の信頼性を計算するためのサブセット224
- ソフト判定を計算するためのサブセット226を含む。

【0112】系統的重畳コードの復号は、ソフト判定復号アルゴリズムを適用することから成る。説明を簡単にするために、一例として、7/8なるレート of の系統的重畳コードについて考える。表1に示すインデックス1を持つロウについて具体的に説明すると、一連の生成コードは、以下のシンボルから形成され、これらが送信端から送信される (説明を簡単にするために、以下では、ロウのインデックス (ここでは、 $i=1$) は省略される)。

7N+4個のデータシンボル: $I = I_1, \dots, I_{7N+4}$

N個のコード冗長シンボル:

$R = R_1, \dots, R_N$

4個のトレリス閉鎖シンボル:

$TR = TR_1, TR_2, TR_3, TR_4$

送信チャネルの不完全さのために、受信されたシンボル $D(0) = (D(0)_1, \dots, D(0)_{8N+8})$ は、送信されたシンボルに対して差異を示す。

【0113】受信機端では複合シンボルが受信されるが、これは、チャネルを通った後の送信された16-QAMシンボルに対応する。各複合シンボルは、2個の実数シンボルのシーケンスであるとみなされる。次に、これら実数シンボルが、送信機端に定義されるマトリックスと同一構造を持つマトリックスに従ってメモリ20

(図1) 内に格納される。これら受信された実数シンボルは、送信機端で使用されたのと同じ順番を守りながらマトリックスに配列される。ビット i の復号プロセス

は反復的であるが、これは、マトリックスのロウが最初に復号され、次に、カラムが復号されることを意味する。各反復において復号されるべきロウあるいはカラムの数は、表1の場合(冗長の符号化を行なわない場合)は、 $7N+4$ に等しく、冗長が符号化される場合(表2)は、 $8N+8$ に等しい。

【0114】最初に、マトリックスのロウに対する v 番目の反復における、つまり、実数シンボルのブロック： $D(v-1) = (D(v-1)_1, \dots, D(v-1)_{8N+8})$ の反復的復号について考える。インデックス v は、反復の回数に対応する整数である。復号によって $8N+8$ 個の実数シンボル(ソフト出力)のブロック： $D(v) = (D(v)_1, \dots, D(v)_{8N+8})$ が生成され、これが、それぞれ、次の反復と、送信された4-AMシンボル($I_1, \dots, I_{7N+4}, R_1, \dots, R_N, TR_1, \dots, TR_4$)の第一の分割レベルのビット(LSBビット)である送信されたビット $i = (i_1, \dots, i_{8N+8})$ に対応するビット $\hat{i}(v) = (\hat{i}(v)_1, \dots, \hat{i}(v)_{8N+8})$ の最適判定に加えられる。ここでは、 $7N+4$ 個のシンボル： $D(v-1)_1, \dots, D(v-1)_{7N+4}$ がデータシンボルに対応し、 N 個のシンボル： $D(v-1)_{7N+5}, \dots, D(v-1)_{8N+4}$ が $7/8$ なるレートのコード冗長シンボルに対応し、4個のシンボル： $D(v-1)_{8N+5}, \dots, D(v-1)_{8N+8}$ がトレリスを閉鎖するための冗長シンボルに対応するものと想定する。これらシンボルが以下の順番で処理される：

【0115】

【数10】

$$\begin{aligned} & D_1^{(v-1)}, D_7^{(v-1)}, D_{7N+5}^{(v-1)}; \\ & D_8^{(v-1)}, D_{14}^{(v-1)}, D_{7N+6}^{(v-1)}; \\ & \dots \\ & D_{7N-6}^{(v-1)}, \dots, D_{7N}^{(v-1)}, D_{8N+4}^{(v-1)}; \\ & D_{7N+1}^{(v-1)}, \dots, D_{7N+4}^{(v-1)}, D_{8N+5}^{(v-1)}, \dots, D_{8N+8}^{(v-1)}. \end{aligned}$$

復号は、6つのステップにて遂行される。以下の説明は、 $D(v-1)$ から $D(v)$ を導出する方法について示す。

【0116】第1のステップは、距離(metrics)の計算に関する。距離を計算するためのサブセット220(図13)は、8個のシンボルからなる各ブロックについて(1番目のブロック、 $l=0, \dots, N-1$)($D(v-1)_{7l+1}, \dots, D(v-1)_{7l+7}, D(v-1)_{7N+5l+1}$ について)、これを検出すると共に各8-Dサブセット $S_{i,j}$ における距離の計算を遂行し、さらに、生成コードのロウあるいはカラムの符号化の終端においてトレリスを閉鎖する遷移に対応するブロック($D(v-1)_{7N+1}, D(v-1)_{7N+4}, D(v-1)_{8N+5}, D(v-1)_{8N+8}$)についても、これ

を検出すると共に距離の計算を遂行する。簡単のため

に、これら8-Dブロックは、 (r_1, \dots, r_8) として示される。距離の計算は、さまざまなステップによって遂行される。

【0117】最初に距離が1-D群(constellation)において計算される。 $k=1, \dots, 7$ (ブロックの最初の7個のシンボル)に対して、サブセット $B_0 = \{3, -1\}$ 、および $B_1 = \{1, -3\}$ のそれぞれにおける最も近い4-AMシンボルが検出され、それぞれ、これに対応する距離： $M_{k,0}$ 、および $M_{k,1}$ が計算される。ここで、 $M_{k,i}$ は、 r_k と、 B_i における最も近いシンボルとの間の二乗ユークリッド距離に等しい。

【0118】 $k=8$ に対して、各サブセット C_i における、シンボル r_8 の、それぞれ、 $C_i=0, 1, 2, 3$ に対応する距離 $M_{8,0}, M_{8,1}, M_{8,2}, M_{8,3}$ が計算される。ここで、 C_i は、4-AM群に割当てられた2ビットの10進形式で表現された値 i に対応する4-AMポイントを含む。

【0119】その後、これらの距離が、2-D群において計算される。最初の3個の2-Dブロック(シンボルペア)、つまり、 $(r_1, r_2), (r_3, r_4), (r_5, r_6)$ に対して、距離 $M_{k,l}^{(2)}$ ($k=1, 3, 5$)が、4個の2-Dサブセット $E^{(2)}_l = B_i B_{i'}$ において検出および計算される。ここで、 $l=2i+i'$ 、 $i=0, 1, i'=0, 1$

2-D距離(metric)は、2-Dブロックの2個のシンボルの2個の1-D距離の総和として定義される。 $E^{(2)}_l$ における (r_k, r_{k+1}) の距離が、 $k=1, 3, 5$ に對以下のように得られる：

$$M_{k,l}^{(2)} = M_{k,i} + M_{k+1,i'} \quad (l=0, 1, 2, 3 \text{ に対して})$$

ここで、 $l=2i+i'$ である。

【0120】最後の2-Dブロック r_7, r_8 に対して、距離 $M_{7,l,j}^{(2)}$ が、8個の2-Dサブセット $E^{(2)}_{l,j} = B_i C_{2j+i'}$ において、 $i=0, 1, i'=0, 1, j=0, 1$ に対して、以下のように計算される。ここで、 $l=2i+i'$ である。：

$$M_{7,l,j}^{(2)} = M_{7,i} + M_{8,2j+i'}$$

その後、この距離(metrics)が4-D群において計算される。4-Dブロック $(r_1 r_2 r_3 r_4)$ の距離 $M_{1,l}^{(4)}$ が、8個の4-Dサブセット $E^{(4)}_l$ ($l=0, 1, \dots, 7$)に対して以下のように検出および計算される：

【0121】

【数11】

$$\begin{aligned}
E_0^{(4)} &= E_0^{(2)} E_0^{(2)} \cup E_3^{(2)} E_3^{(2)} \\
E_1^{(4)} &= E_0^{(2)} E_1^{(2)} \cup E_3^{(2)} E_2^{(2)} \\
E_2^{(4)} &= E_0^{(2)} E_2^{(2)} \cup E_3^{(2)} E_1^{(2)} \\
E_3^{(4)} &= E_0^{(2)} E_3^{(2)} \cup E_3^{(2)} E_0^{(2)} \\
E_4^{(4)} &= E_1^{(2)} E_0^{(2)} \cup E_2^{(2)} E_3^{(2)} \\
E_5^{(4)} &= E_1^{(2)} E_1^{(2)} \cup E_2^{(2)} E_2^{(2)} \\
E_6^{(4)} &= E_1^{(2)} E_2^{(2)} \cup E_2^{(2)} E_1^{(2)} \\
E_7^{(4)} &= E_1^{(2)} E_3^{(2)} \cup E_2^{(2)} E_0^{(2)}
\end{aligned}$$

各サブセットに対して、2個の4-D距離が計算され、小さい方の距離が採用される。例えば、 $E_0^{(4)}$ に対して以下が計算される： $E_0^{(2)}E_3^{(2)}$ における距離 $M^{(2)}_{1,3} + M^{(2)}_{3,3}$ ； $E_3^{(2)}E_0^{(2)}$ における距離 $M^{(2)}_{1,3} + M^{(2)}_{3,3}$ ； $E_0^{(4)}$ における $(r_1 r_2 r_3 r_4)$ の距離 $M^{(4)}_{1,0}$ ：これは、 $\min [M^{(2)}_{1,0} + M^{(2)}_{3,0}, M^{(2)}_{1,3} + M^{(2)}_{3,3}]$ に等しい。

【0122】各ブロック $(r_5 r_6 r_7 r_8)$ に対して、サブセット $E_{1,j}^{(4)}$ における距離 $M^{(4)}_{5,1,j}$ が、 $1=0, 1, \dots, 7$ 、および $j=0, 1$ に対して検出および計算される。サブセット $E_{1,j}^{(4)}$ は、 $E_1^{(4)}$ と、最後のシンボルが $j=0$ に対しては正であり、 $j=1$ に対しては負である点を除いて類似に定義される。例えば、 $E_{0,j}^{(4)} = E_0^{(2)}E_0^{(2)} \cup E_3^{(2)}E_3^{(2)}$ として定義される。このために、第2の2-Dサブセットのインデックス k を、 k, j によって置き換えることで十分である。 $E_{0,j}^{(4)}$ における距離は： $M^{(4)}_{5,0,j} = \min [M^{(2)}_{5,0} + M^{(2)}_{7,0,j}, M^{(2)}_{5,3} + M^{(2)}_{7,3,j}]$ となる。

【0123】最後に、距離が16個の8-Dサブセットにおいて計算される。偶数の i に対するサブセット（部分集合） $S_{i,j}$ は、おのおのが2個の4-Dサブセットの連接である4個の8-Dサブセット（部分集合）の和集合である：

$S_{i,j} = E_{1,j}^{(4)} E_{1,j}^{(4)}$ 、ここで、 $i_2=0, 1, i_3=0, 1$ そして：

【0124】

【数12】

$$\begin{aligned}
i &= 8Y_3 + 4Y_2 + 2Y_1 ; \\
1 &= 4i_2 + 2i_3 + i_4 ; \\
1' &= 4i_6 + 2i_7 + i_8 ; \\
i_4 &= i_2 \oplus i_3 \oplus Y_1 \\
i_6 &= i_2 \oplus Y_2 \\
i_7 &= i_3 \oplus Y_3 \\
i_8 &= i_2 \oplus i_3 \oplus i_4 \oplus i_6 \oplus i_7 .
\end{aligned}$$

10

$S_{i,j}$ における r_1, \dots, r_8 の距離の計算には、4個の8-Dサブセット $E_{1,j}^{(4)} E_{1,j}^{(4)}$ における距離を計算すること、つまり、 $E_{1,j}^{(4)}$ における $(r_1 r_2 r_3 r_4)$ の距離と $E_{1,j}^{(4)}$ における $(r_5 r_6 r_7 r_8)$ の距離との総和を計算することが必要となる。4個の距離の内の最小によって、 $S_{i,j}$ における8-D距離 $M^{(8)}_{i,j}$ が与えられる。

【0125】各8-Dブロックに対して、各サブセット $S_{i,j}$ における距離が検出および計算され、これらの距離がビタビ復号器に供給される。これら距離は、遷移 $1, 2, \dots, N+1$ の際のトレリスブランチの距離でもある。

【0126】これもやはり第1の段において、復号の第2のステップが遂行される。サブセット222は、受信されたシンボルシーケンスの従来のビタビ復号を遂行する。このようにして、以下のようなロウの復号された最適シーケンス（ハード判定）が得られる：

30 $\hat{I}^{(v)} = \hat{I}^{(v)}_1, \dots, \hat{I}^{(v)}_{7N+4}, \hat{I}^{(v)}_{7N+5}, \dots, \hat{I}^{(v)}_{8N+8}$ および復号されたLSBビットシーケンス： $\hat{i}^{(v)} = \hat{i}^{(v)}_1, \dots, \hat{i}^{(v)}_{8N+8}$ および8-Dサブセットに対応するシーケンス

復号された状態に対応する状態のシーケンスは： $\hat{\sigma}^{(i)} = \hat{\sigma}^{(i)}_0, \dots, \hat{\sigma}^{(i)}_{N+1}$ である。これは、 $N+1$ 個の遷移あるいはブランチに対応する。ここで、 $\hat{\sigma}^{(i)}_0 = \hat{\sigma}^{(i)}_{N+1} = 0$ は、初期および最終の0の状態である。状態距離 $M^{(i)}_j(\sigma)$ のベクトルが、各状態 $\sigma = 0, \dots, 7$ 、および各瞬間 $j = 1, \dots, N+1$ について格納される。最終的に、テーブルにおいて生き残った経路が格納される。ビタビ復号器222によって生成されるこれら全ての結果は信頼性を計算するために使用される。

【0127】第3のステップにおいて、復号されたシーケンス $\hat{I}^{(v)}$ の、各シンボル $\hat{I}^{(v)}_k$ 、あるいは各ビット $\hat{i}^{(v)}_k$ の信頼性が、 $k = 1, \dots, 8N+8$ に対して、サブセット224において計算される。この信頼性は、復号されたシンボルの品質（正確さの確率）の尺度である。この信頼性は以下のように書き表すことができる：

50 $F^{(i)} = F^{(i)}_1, \dots, F^{(i)}_{8N+8}$

k 番目のオーダのシンボルの信頼性は、以下のように書き表すことができる：

【0 1 2 8】

【数 1 3】

$$F_k^{(v)} = \log \left[\frac{(\sum_{\hat{I}(v)} \text{Prob}(D^{(v-1)} | C))}{\text{Prob}(D^{(v-1)} | \hat{I}(v))} \right]$$

*

$$F_k^{(v)} = \log \left[\sum_{\hat{I}(v)} c(\hat{I}(v)) \exp \frac{d^2(D^{(v-1)}, \hat{I}(v)) - d^2(D^{(v-1)}, C)}{N_0} \right]$$

総和の主項を取ることによって、以下のように書き替えることが可能である：

※ 【0 1 3 0】

※ 【数 1 5】

$$F_k^{(v)} \approx \min_{\hat{I}(v)} \left[d^2(C, D^{(v-1)}) - d^2(\hat{I}(v), D^{(v-1)}) \right]$$

この最小は、シンボル $\hat{I}(v)_k$ に近い復号された 4-A M シンボルを生成するエラー経路との最大の対応を与える。この信頼性の定義は、ビット $\hat{i}(v)_k$ の信頼性の定義でもある。事実、 $\hat{I}(v)_k$ に近いシンボルは、 $\hat{i}(v)_k$ の補数であるビット i に対応する。

【0 1 3 1】信頼性の計算は、以下のアルゴリズムに従って行なわれる。

【0 1 3 2】ビタビ復号器によって復号されたシーケンス $\hat{I}(v)_k$ は、 $N+1$ なる長さを持つトレリス内のある経路に対応する。判定 $\hat{I}(v)_k$ の信頼性 $F_k^{(v)}$ を、 $k=1, \dots, 8N+8$ に対して計算するための計算アルゴリズムは以下の通りである：判定 $\hat{I}(v)_k$ の信頼性 ($k=1, \dots, 8N+8$)、あるいはビット $\hat{i}(v)_k$ の信頼性 ($k=1, \dots, 8N+8$) の計算は、2つのステップにて遂行される。第1のステップは、信頼性を、並列な遷移に対応するエラー経路と比較して計算することから成り、第2のステップは、信頼性を、トレリスのエラー経路と比較して計算することから成る。最終的な信頼性は、この2つの内の低い方の値として決定される。

【0 1 3 3】最初に、並列なブランチに起因する信頼性 F_P が計算される。このプロセスは、シンボルシーケンス $D^{(v-1)}_k$ 、および $I(v)_k$ に関して、ブロック毎に遂行★

20 ★される。各処理の結果として、1 番目の遷移 ($l=0, \dots, N-1$) に対応する 8 個のシンボル ($D^{(v-1)}_{7l+1}, \dots, D^{(v-1)}_{7l+7}, D^{(v-1)}_{7N+5+1}$) に起因する信頼性と、($N+1$) 番目の遷移 (つまりトレリスを閉鎖する遷移) に対応する 8 個のシンボル ($D^{(v-1)}_{7N+1}, \dots, D^{(v-1)}_{7N+4}, D^{(v-1)}_{8N+5}, \dots, D^{(v-1)}_{8N+8}$) に起因する信頼性を計算することが可能になる。トレリスの並列なブランチが、8-D サブセット $S_{i,j}$ に割当てられるために、この信頼性は、 $\hat{I}(v)_k$ に対応する復号されたブロックを含む 8-D サブセットにおける判定 $\hat{I}(v)_k$ の信頼性でもある (復号された 8-D サブセットのシーケンスはビタビ復号器によって生成される)。

【0 1 3 4】 $S_i(r_1, \dots, r_8)$ は、例えば、ブロック ($D^{(v-1)}_1, \dots, D^{(v-1)}_7, D^{(v-1)}_{7N+5}$) であり、 $S_{i,j}$ は、検出された 4-AM シンボル (d_1, \dots, d_8) = ($\hat{I}(v)_1, \dots, \hat{I}(v)_7, \hat{I}(v)_{7N+5}$) に対応する復号された 8-D サブセットであるために、 $S_{i,j}$ における信頼性 r_k ($k=1, \dots, 8$) 計算は、煮詰めると、サブセット $S_{0,j'}$ における以下の量の信頼性の計算に帰着する。

40 【0 1 3 5】

【数 1 6】

$$(r_1, r_2, r_3, (1-2Y_1)r_4, r_5, (1-2Y_2)r_6, (1-2Y_3)r_7, (1-2(Y_1 \oplus Y_2 \oplus Y_3))r_8)$$

ここで $j' = j \oplus Y_1 \oplus Y_2 \oplus Y_3$ 、であり、 $i = 8Y_3 + 4Y_2 + 2Y_1$ 。

こうして、全てのケースにおいて、これは、煮詰めると、 $S_{0,0}$ あるいは $S_{0,1}$ の信頼性の計算に帰着し、これは、アルゴリズムをより単純にする。

【0 1 3 6】図 1 4 は、サブセット S_{00} と S_{01} のトレリ

スを表す。ここで、トレリス内の各経路は、 $B_{i1}B_{i2} \dots B_{i7}C_{2j8+i8}$ なる形式を持つサブセットに対応する。

50 【0 1 3 7】次に、 $S_{0,0}$ 、つまり、 $S_{i,j} = S_{0,0}$ に対

して、判定 d_k ($k=1, \dots, 8$) 信頼性を計算するためのアルゴリズムについて説明する。

【0138】最初に、2-D距離の計算が以下のように行なわれる：

【0139】

【数17】

$$M_{i,j}(1) = M_{1,j} + M_{2,i} \quad i=0, 1; j=0, 1$$

$$M_{i,j}(2) = M_{2,j} + M_{4,i} \quad i=0, 1; j=0, 1$$

$$M_{i,j}(3) = M_{5,i} + M_{6,j} \quad i=0, 1; j=0, 1$$

$$M_{i,j}(4) = M_{7,i} + M_{8,j} \quad i=0, 1; j=0, 1$$

ここで、 $M_{k,i}$ は、 B_i における r_k ($k=1, \dots, 7$) の距離であり； $M_{8,i}$ は、 C_i における r_8 の距離である。

【0140】信頼性の計算が $S_{0,1}$ に対して遂行される場合は、 $M_{8,i}$ が、 C_2 および C_3 ($i=2, 3$) について計算される。

【0141】こうして、 (r_1, \dots, r_8) からの最も近い経路が以下のように $S_{0,0}$ について求められる： $S_{0,0}$ のトレリスの上側部分と下側部分とは別々に計算される。上側トレリスにおいては、 $k=1, 2, 3$ に対する $M(0, k) = \min(M_{0,0}(k), M_{1,1}(k))$ と、 $M(0, 4) = \min(M_{0,0}(4), M_{1,1}(4))$ が計算される。

【0142】最小の値を持つビット (i_1, \dots, i_8) が、 $i_1=i_2, i_3=i_4, i_5=i_6, i_7=i_8$ として、テーブル [$i(0, 1) \dots i(0, 8)$] 内に格納される。そして：

【0143】

【数18】

$$MET(0) = \sum_{k=1}^4 M(0, k)$$

が計算される。

【0144】下側トレリスにおいては、 $k=1, 2, 3$ に対する $M(1, k) = \min(M_{0,1}(k), M_{1,0}(k))$ と、 $M(1, 4) = \min(M_{0,1}(4), M_{1,0}(4))$ に対して計算され、そして：

【0145】

【数19】

$$MET(1) = \sum_{k=1}^4 M(1, k)$$

が計算される。

【0146】 [$i(1, 1), \dots, i(1, 8)$] における最小に対応するビット i_1, \dots, i_8 が、 $i_1=i_2, i_3=i_4, i_5=i_6, i_7=i_8$ として、格

納される。

【0147】シーケンス [$i(0, 1), \dots, i(0, 8)$]、および [$i(1, 1), \dots, i(1, 8)$] が2つの有効(valid)なシーケンスであるか否か検証(verify)される。パリティビット $p(0)$ 、および $p(1)$ が、それぞれ、シーケンス [$i(0, 1), i(0, 3), i(0, 5), i(0, 7)$]、および [$i(1, 1), i(1, 3), i(1, 5), i(1, 7)$] に対して計算される。 $k=0, 1$ に対して、 $p(k)=0$ の場合は、対応するシーケンスは、トレリス(すなわち $S_{0,0}$)に属し、そうでない場合は、そのシーケンスは、修正される。

【0148】2-D距離の差が、以下のように計算される：

【0149】

【数20】

$$\delta(0, k) = |M_{0,0}(k) - M_{1,1}(k)|, \quad k=1, 2, 3$$

$$\delta(0, k) = |M_{0,0}(k) - M_{1,1}(k)|, \quad k=4$$

$$\delta(1, k) = |M_{0,1}(k) - M_{1,0}(k)|, \quad k=1, 2, 3$$

$$\delta(1, k) = |M_{0,1}(k) - M_{1,0}(k)|, \quad k=4$$

次に、4個の距離の最小が、2つの上側トレリス ($q=0$)、および下側トレリス ($q=1$) のそれぞれに対して計算される：

【0150】

30 【数21】

$$\Delta(q) = \min_{k=1, \dots, 4} \delta(q, k), \quad q=0, 1$$

$k_{\min}(q)$ における最小を与えるインデックス k が格納され、第二の最小 $\Delta_2(q)$ が計算される。

【0151】パリティビット $p(q)=1$ である場合は、ビット ($i(q, 2k_{\min}(q)-1), i(q, 2k_{\min}(q))$) の補数が取られ、 $\Delta(q)$ の距離MET(q)が増分される。この動作が、 $q=0$ (上側トレリス)と、 $q=1$ (下側トレリス)に対して遂行される。

40

【0152】次に、 $\Delta_{\min} = |MET(0) - MET(1)|$ が計算され、MET(0)とMET(1)の内のより低い値を与えるインデックス c (0あるいは1)が格納される。

【0153】 $S_{0,0}$ において (d_1, \dots, d_8) の信頼性 $F_P(1), \dots, F_P(8)$ が以下のように初期化される：

【0154】

【数22】

$$F P(2K-1) = F P(2K) = \delta(c, k) + (1-2p(c)) \Delta(c) \quad \text{ここで } K \neq K_{\min}(c)$$

$$F P(2K-1) = F P(2K) = \Delta_2(c) + (1-2p(c)) \Delta(c) \quad \text{ここで } K = K_{\min}(c)$$

ここで、 $F P(k)$ は、 $S_{0,0}$ における d_k の信頼性である。

【0155】その後、信頼性 $F P(k)$ ($k=1, \dots, 8$) が以下のように更新される： $k=1, \dots, 8$ に対して：

$$F P(k) = \min[F P(k), \Delta_{\min}]$$

ここで、 $i(c, k) \neq i(1-c, k)$

$k=1, 2, 3, 4$ に対して：

$i(0, 2k-1) = i(1, 2k-1)$ の場合は

$$F P(2k-1) = \min(F P(2k-1), A_k)$$

$i(0, 2k) = i(1, 2k)$ の場合は

$$F P(2k-1) = \min(F P(2k), A_k)$$

ここで $k \neq k_{\min}(1-c)$ の場合は

$$A_k = \Delta_{\min} + \delta(1-c, k) + [1-2p(1-c)] \Delta(1-c)$$

$k = k_{\min}(1-c)$ の場合は

$$A_k = \Delta_{\min} + \Delta_2(1-c) + [1-2p(1-c)] \Delta(1-c)$$

次に、トレリスのエラー経路に起因する信頼性が決定される(図15)。

【0156】ロウの復号について考える(ステップ300)。

【0157】最初に、 $\hat{I}^{(v)}_k, M^{(v)}_k(\sigma), D^{(v-1)}_k$ から開始される。

【0158】ここで、 $\hat{I}^{(v)}_k$ は、ビタビ復号器によって $k=1, \dots, 8N+8$ に対して復号された4-AMシンボルのシーケンスであり； $M^{(v)}_k(\sigma)$ は、瞬間 k (k 番目の遷移) ($k=1, \dots, N+1$)における状態 σ の距離であり； $D^{(v-1)}_k$ は、 $k=1, \dots, 8N+8$ に対する(先行する反復復号器によって生成された)入力シンボルである。

【0159】次に(図15のステップ301)、信頼性 $F(v) = (F(v)_k, k=1, \dots, 8N+8)$ から信頼性 $F P(v) = (F P(v)_k, k=1, \dots, 8N+8)$ のアレイが計算される。

【0160】ここで、 $F P(v)_k$ は、並列な遷移に起因する $\hat{I}^{(v)}_k$ の信頼性である。

【0161】次に、 $k=N+1, \dots, 2$ (ここで k は一つの瞬間、つまり、遷移のインデックスに対応)に対して以下の動作を遂行する(ステップ302)：

a) 復号経路における k 番目の遷移の後の符号器の状態 $\hat{\sigma}^{(v)}_k$ に基づいて、復号経路の($k-1$)番目の遷移の後に状態 $\hat{\sigma}^{(v)}_{k-1}$ が決定され、 $\hat{\sigma}^{(v)}_k$ の7個の他の先駆状態、つまり、 $\sigma^{(v)}_{j,k-1}$ ($j=1, \dots, 7$)についても決定される(ステップ304)。この目的のために、 $\hat{\sigma}^{(v)}_k$ の値が、 k を、1から $N+1$ に変動す

ることで、ビタビ復号器によって生成され、 $\sigma^{(v)}_{j,k-1}$ の値がメモリ内に格納される。

【0162】ビタビ復号器内で、 $k=2, \dots, N+1$ に対して計算され、これに格納され、

【0163】

10 【数23】

$$M^{(v)}_{k-1}(\sigma^{(v)}_{1,k-1})$$

$$M^{(v)}_{k-1}(\sigma^{(v)}_{2,k-1})$$

.....

$$M^{(v)}_{k-1}(\sigma^{(v)}_{7,k-1})$$

$$M^{(v)}_k(\hat{\sigma}^{(v)}_k)$$

20

として命名される、状態 $\sigma^{(v)}_{1,k-1}, \sigma^{(v)}_{2,k-1}, \dots, \sigma^{(v)}_{7,k-1}$ および $\hat{\sigma}^{(v)}_k$ の距離に基づいて、状態 $\hat{\sigma}^{(v)}_k$ の7個の累積距離(cumulated metrics)が、7個の状態 $\sigma^{(v)}_{j,k-1}$ ($j=1, \dots, 7$) から来る7つの経路に対して計算される。これら7つの累積距離は：

$$\delta(j) = M^{(v)}_{k-1}(\sigma^{(v)}_{j,k-1}) + MET(\sigma^{(v)}_{j,k-1} \rightarrow \hat{\sigma}^{(v)}_k)$$

である。ここで、 $MET(\sigma^{(v)}_{j,k-1} \rightarrow \hat{\sigma}^{(v)}_k)$ は、状態 $\hat{\sigma}^{(v)}_{j,k-1}$ から状態 $\hat{\sigma}^{(v)}_k$ への遷移の距離であり、これは、この遷移に割当てられた8-Dサブセットの距離でもある。これら距離が、ビタビ復号の際にメモリ内に格納される(ステップ306)。

【0164】最適距離と各累積距離との間の差が計算される：

$$\Delta(j) = \delta(j) - M^{(v)}_k(\hat{\sigma}^{(v)}_k), \quad (j=1, \dots, 7)$$

ここで、 $M^{(v)}_k(\hat{\sigma}^{(v)}_k)$ は、状態 $\hat{\sigma}^{(v)}_k$ の最適距離である。

【0165】その後、状態 $\hat{\sigma}^{(v)}_k$ への7個の生き残った先駆状態が調べられる。 j 番目オーダの生き残った状態の、 $\hat{\sigma}^{(v)}_k$ から $\sigma^{(v)}_{j,k-1}$ へのステップバックが行なわれる。その後、状態 $\sigma^{(v)}_{j,k-1}$ の j 番目の各生き残った状態が、初期状態 $\sigma=0$ に至まで調べられる。これは、遷移のインデックスを k から1まで変動させることからなる。

【0166】複雑さを減らすために、状態 $\sigma^{(v)}_{j,k-1}$ の7個の生き残りを、瞬間1の代わりに、瞬間 $k-L$ における生き残りの状態まで調べることで、遷移の数を固定させることも可能である。 $L=3$ とした場合の性能の劣化は無視できる程度である。 j 番目オーダの生き残りの

50

k' 番目の遷移に割当てられた 8 個の 4-AM シンボルが、8 個の復号されたシンボルと、 k' について、比較される。ここで、 k' は、前者の場合は、 k から 1 まで変化され、簡素化した第二の場合は、 k から $(k-L)$ まで変化される。復号されたシンボルと、同一ランクの生き残り j の k' 番目の遷移のシンボルとが隣接する場合は、復号されたシンボルの信頼性が、 $\Delta(j)$ と、これが、この信頼性よりも低い場合は、置換される（ステップ 3 1 2）。

【0 1 6 7】最小に対応する $\hat{I}(v)_k$ に隣接するシンボル $\sim I(v)_k$ のテーブルを、 k を、1 から $8N+8$ までで変化させることで、計算する。1 つのシンボルの信頼性を更新する度に、その隣接シンボルも、これを、生き残り j の k' 番目のオーダの遷移と関連する対応する 4-AM シンボルと置換することによって、更新される。

【0 1 6 8】最後に、復号されたシンボル $\hat{I}(v)_k$ の信頼性 $F(v)_k$ と、隣接する対応するシンボル $\sim I(v)_k$ の信頼性 $F(v)_k$ が、 $k=1, \dots, 8N+8$ に対して、得られる（ステップ 3 2 6）。

【0 1 6 9】その後、復号の第四のステップにおいて、この信頼性が、ソフト判定 $\sim D(v)_k$ ($k=1, \dots, 8N+8$) を計算できるように正規化される。正規化された信頼性 $F_{\text{norm}}(v)_k$ を計算するには、3 つの方法のいずれかを使用することが可能である。つまり：

$$F_{\text{norm}}(v)_k = F(v)_k / \sqrt{F}$$

あるいは

$$F_{\text{norm}}(v)_k = F(v)_k + (1 - \sqrt{F})$$

あるいは

$$F_{\text{norm}}(v)_k = (F(v)_k / \beta_1) + \beta_2$$

のいずれかが計算される。

【0 1 7 0】ここで、 \sqrt{F} は、全マトリックスについて計算された信頼性 $F(v)_k$ の平均に等しく、 β_1 および β_2 は、 $\sqrt{F} = \beta_1 (1 - \beta_2)$ を検証する定数である。

【0 1 7 1】その後、第五のステップにおいて、重みなしのソフト判定が計算される（図 1 3 のサブセット 2 2 6）。

【0 1 7 2】ソフト判定 $\sim D(v)_k$ は、ビタビ復号器によって与えられた最適判定 $\hat{I}(v)_k$ 、その正規化された信頼性 $F_{\text{norm}}(v)_k$ 、および、前に得られた 4-AM シンボル $\sim I(v)_k$ のシーケンスに基づいて計算される。この 4-AM シンボル $\sim I(v)_k$ は、シンボル $\hat{I}(v)_k$ に隣接するシンボルである。つまり、 $|\sim I(v)_k - \hat{I}(v)_k| = 2$ である。これは、シンボル $\hat{I}(v)_k$ の後に発生する可能性が最も高い 4-AM シンボルに対応する。シンボル $\sim D(v)_k$ は、以下によって与えられる：

$$\sim D(v)_k = 1/2 (\hat{I}(v)_k + \sim I(v)_k) + \text{sgn}(\hat{I}(v)_k - \sim I(v)_k) F_{\text{norm}}(v)_k$$

ここで、関数 sgn 、以下によって定義される：

$$\text{sgn}(x) = \begin{cases} x > 0 \text{ の場合は } 1 \\ x = 0 \text{ の場合は } 0 \\ x < 0 \text{ の場合は } -1 \end{cases}$$

こうして、シンボル $\sim D(v)_k$ は、 $\hat{I}(v)_k$ と $\sim I(v)_k$ の間の判定閾値から距離 $F_{\text{norm}}(v)_k$ の所に位置するシンボルに対応する。E ($F_{\text{norm}}(v)_k$) = 1 であるために、シンボル $\sim D(v)_k$ は、4-AM シンボルの中心付近に来る。

【0 1 7 3】第 6 のステップにおいて、性能を向上させるために、チャネルを通じて受信され、メモリ 2 0 内に格納されたシンボル $D(0)_k$ を用いて、 $\sim D(v)_k$ に重みが付けられ、次に、これを用いて $D(v)_k$ が以下のように計算される：

$$D(v)_k = \alpha_v \sim D(v)_k + (1 - \alpha_v) D(0)_k$$

ここで、 α_v は、重み係数である。例えば、 $\alpha_1 = 0.6$ ； $\alpha_2 = 0.9$ ；そして v が 1 より大きな場合は、 $\alpha_v = 1$ とされる。目的は、第 1 の反復の際のエラーの伝搬の効果を低減することにある。ステップ 5 とステップ 6 は、サブセット 2 2 6 内で一緒に遂行される。

【0 1 7 4】次に、第 2 の段（経路 2）におけるビット j を復号する動作について説明する。ビット j 、つまり、4-AM シンボルの MSB ビットは、生成コードのマトリックスの各ロウおよび各カラムにパリティコードを適用することによって符号化され、これは、トレリスを閉鎖するときの実現される。これが、ビット i の場合と同様に、反復的に復号される。

【0 1 7 5】図 1 6 は、送信された 4-AM シンボルの MSB ビットのパリティコードを復号するための復号手段 2 2 1 を含む段の一般回路図である。この段は：

30 - パリティコードを復号し、ハード判定を与えるための復号器 3 2 2

- 判定の信頼性を計算するためのサブセット 3 2 4

- ソフト判定を計算するためのサブセット 3 2 6

を含む。

【0 1 7 6】第一の反復 ($m=1$) に対しては、以下のシンボル $x(0)_k$ が、チャネルの出力上に受信される：

$$D(0)_1, \dots, D(0)_{7N+4}, D(0)_{8N+5}, \dots, D(0)_{8N+7}$$

図 1 7 は、パリティコードにて符号化された $7N+7$ 個のビット ($j_1, \dots, j_{7N+4}, j_{8N+5}, j_{8N+6}, j_{8N+7}$) をトレリス表現にて示す。このトレリスにおいて、ポイント I からポイント F への任意の経路は、 $7N+7$ の長さを持つあるコード語に対応する。 m 番目のランクの反復に対する $7N+7$ 個のシンボルのブロック $x^{(m-1)}_k$ の復号は、3 つのステップによって遂行される。

【0 1 7 7】第 1 のステップ（ステップ 3 2 2）において、 $x^{(m-1)}_k$ にいつのハード判定復号が行なわれる。サブセット B^{ik} において、これらシンボルに、シンボル毎に、閾値判定が適用される。ここで、 ik ($k=1, \dots, 7N+4, 8N+5, 8N+6, 8N+$

50 7) は、第一の段の反復符号化によって推定された収束

後の4-AMシンボルのLSBビットのシーケンスである。この閾値判定によって、第1のビット推定 $\hat{j}^{(m)}_k$ が生成される。ここで、 $\hat{j}^{(m)}_k$ は、 $B \cdot i_k$ における判定が正である場合は0、負である場合は1である。

【0178】その後、各シンボル $x^{(m-1)}_k$ に対して、最初に、サブセット $B \cdot i_k$ におけるこのシンボルから判定閾値 $t^{(m-1)}_k$ までの距離 $d(x^{(m-1)}_k, t^{(m-1)}_k)$ の絶対値が計算される。ここで、 $B_0 = \{3, -1\}$ においては閾値は、+1に等しく、 $B_1 = \{1, -3\}$ においては閾値は、-1に等しい。次に、 k に対して、距離 $d(x^{(m-1)}_k, t^{(m-1)}_k)$ の最小 Δ が計算され、第2の最小 Δ_2 についても計算される。

【0179】最後に、ビット $\hat{j}^{(m)}_k$ のパリティビット*

$$FJ_k^{(n)} = (1-2p)\Delta + d(x_k^{(m-1)}, t_k^{(m-1)})_{k \neq k_{\min}}$$

$$FJ_k^{(m)} = (1-2p)\Delta + \Delta_2, k = k_{\min}$$

その後、信頼性が以下に従って正規化される：

【0183】

【数25】

$$FJ_k^{\text{norm}(m)} = FJ_k^{(m)} / \overline{FJ}$$

ここで、 \overline{FJ} は、信頼性 $FJ^{(m)}_k$ の平均である。 ※

$$x_k^{(n)} = t_k^{(n)} + 2 \operatorname{sgn}(1 - 2\hat{j}_k^{(m)}) x FJ_k^{\text{norm}(m)}$$

ここで、 $\operatorname{sgn}(\cdot)$ は、符号関数である。

【0186】本発明が、7/8なるレート of 的系統的重畳コードの場合について説明された。当業者においては、同一の符号化および復号原理を、 $P/(Q, M)$ なるレート of 的他の系統的重畳コードに対して、次元 Q および2M個の状態を持つ他の多次元振幅変調と結合して、適用することも可能であることを理解できるものである。

【図面の簡単な説明】

【図1】本発明によるベースバンドデジタル伝送システムの略図を示す。

【図2】木(集合)A0のサブセット(部分集合)への分割を示す。

【図3】生成コードを生成するためのロウあるいはカラム符号器の一般回路図を示す。

【図4】冗長シンボルについても符号化を行なうための符号器を含むデジタル伝送システムを示す。

【図5】8-D星座に分割されたツリー構造(部分集合)を示す。

【図6】7/8なるレート of のコードのトレリスを示す。

【図7】4-AM 8-D変調の結合された7/8なるレート of 的系統的重畳符号器の一般回路図を示す。

【図8】4-AMシンボルのビット割当てを示す。

【図9】8-状態コードに対する系統的重畳符号器を示す。

* p が計算される。パリティビットが満足できるものである場合(つまり、 $p=0$ である場合)は、ビット $\hat{j}^{(m)}_k$ のシーケンスは、最適シーケンスである。そうでない場合は、ビット $\hat{j}^{(m)}_{k_{\min}}$ の補数が取られる。ここで、 k_{\min} は、距離 $d(x^{(m-1)}_k, t^{(m-1)}_k)$ の最小 Δ を与えるインデックスである。

【0180】第二のステップにおいて、信頼性の計算が行なわれる(ステップ324)。

【0181】判定 $\hat{j}^{(m)}_k$ の信頼性は、以下によって与えられる：

【0182】

【数24】

※【0184】第三のステップにおいて、ソフト判定 $x^{(m)}_k$ の計算が行なわれる(ステップ326)。この判定は、 $t^{(m)}_k$ 、 $\hat{j}^{(m)}_k$ 、 $FJ^{\text{norm}(m)}_k$ の関数として、以下に従って得られる：

【0185】

【数26】

【図10】16状態コードに対する系統的重畳符号器を示す。

【図11】3/4なるレート of のコードに対する図7に示す符号器の特定の実施例を示す。

【図12】トレリス閉鎖シンボルを計算するための略図を示す。

【図13】本発明による符号化手段の第一の段を示す。

【図14】サブセット $S_{0,0}$ および $S_{0,1}$ のトレリスを示す。

【図15】第一の段において遂行される復号のさまざまなステップの流れ図を示す。

【図16】本発明による第二の段の復号手段を示す。

【図17】パリティコードのトレリスを示す。

40 【符号の説明】

1 経路

2 経路

5 ソース

10 メモリ

12 ロウ符号化手段

13 スイッチ

14 カラム符号化手段

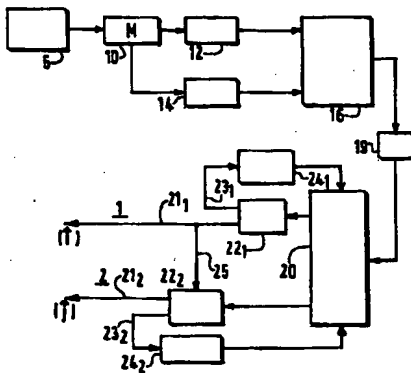
15 スイッチ

16 4-AMデジタル変調シンボル割当て要素

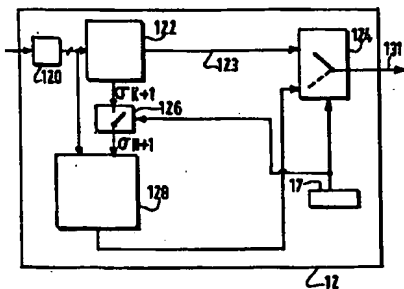
50 17 コントローラ

- 1 9 チャンネル
- 2 0 バッファ手段
- 2 2 1 ビタビ復号器
- 2 1 1 ハード判定出力
- 2 3 1 ソフト判定出力
- 2 4 1 デジタルインタリイバ
- 2 2 2 パリティコード復号器
- 2 1 1 ハード判定出力
- 2 3 2 ソフト判定出力
- 2 4 2 データインタリイバ
- 1 2 0 バッファ

【図 1】

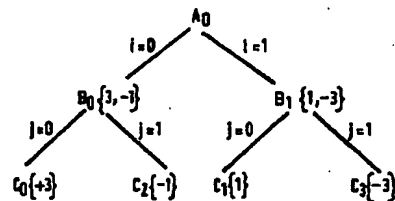


【図 3】

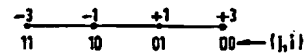


- 1 2 2 系統的重畳ロウ符号器
- 1 2 4 マルチプレクサ
- 1 2 8 トレリス閉鎖手段
- 1 3 1 出力
- 2 1 0 テーブル
- 2 2 0 テーブル (メモリ)
- 2 2 2 遅延セル
- 2 3 0 系統的重畳符号器
- 3 1 0 遅延セル
- 10 3 1 2 加算器セル
- 3 2 2 復号器

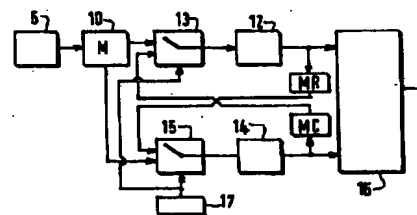
【図 2】



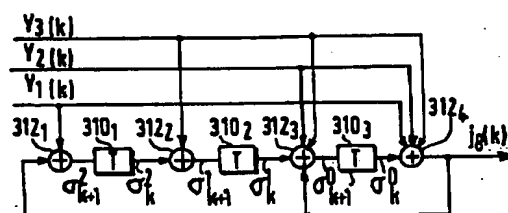
【図 8】



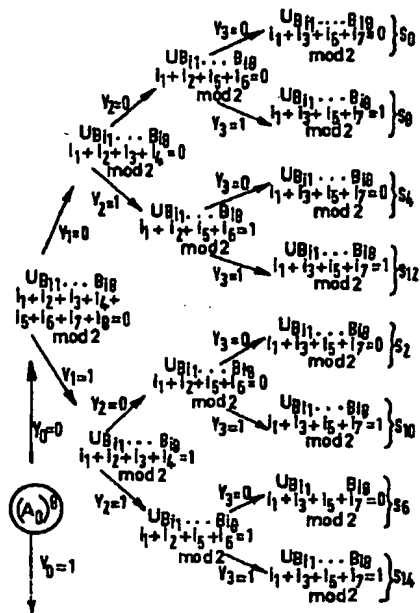
【図 4】



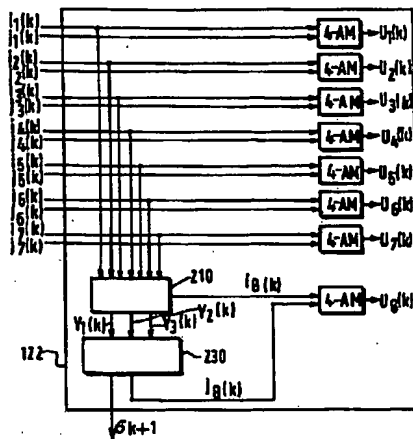
【図 9】



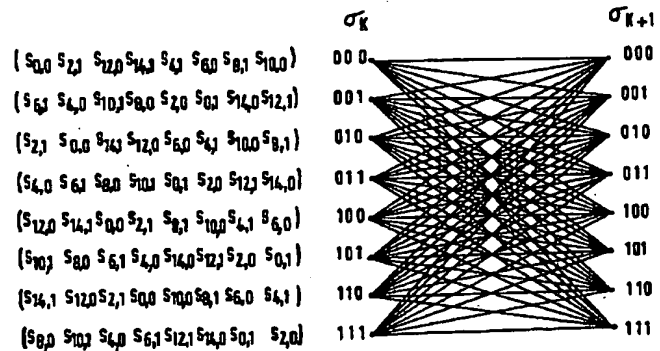
【図 5】



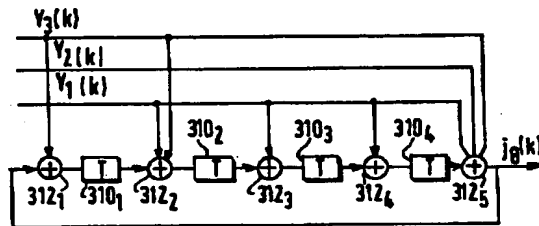
【図 7】



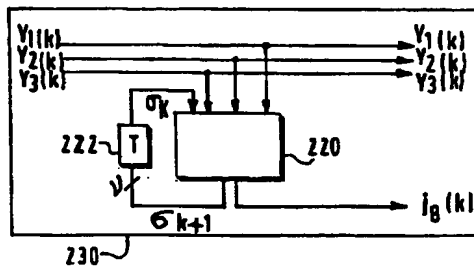
【図 6】



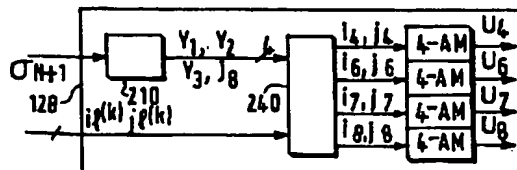
【図 10】



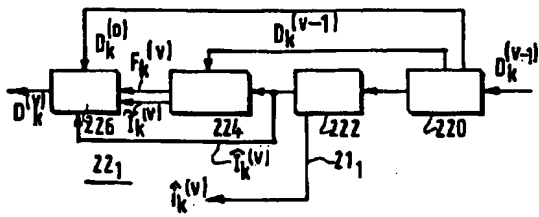
【図 11】



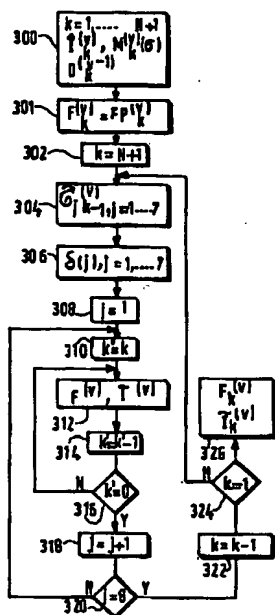
【図 12】



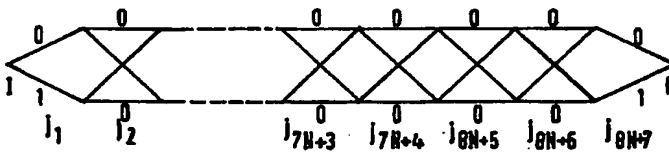
【図 1 3】



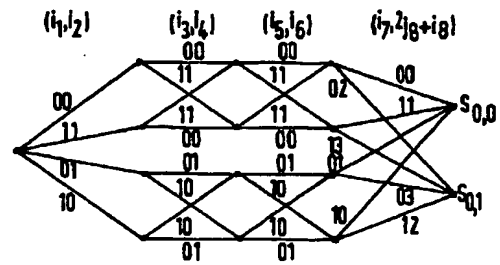
【図 1 5】



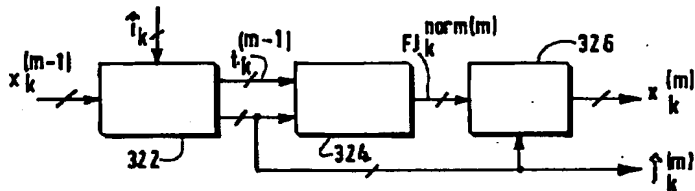
【図 1 7】



【図 1 4】



【図 1 6】



フロントページの続き

(71)出願人 590000248

Groenewoudseweg 1,
5621 BA Eindhoven, Th
e Netherlands

(72)発明者 ダビッド、ジャヌラ

フランス国ダヌマリー、リュ、デ、リラ、
10